(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 3 145 192 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
22.03.2017 Bulletin 2017/12

(51) Int Cl.:
H04N 19/597 *(2014.01)*

(21) Application number: **15306438.1**

(22) Date of filing: **17.09.2015**

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR
Designated Extension States:
BA ME
Designated Validation States:
MA

(71) Applicant: **Thomson Licensing**
92130 Issy-les-Moulineaux (FR)

(72) Inventors:
- **DRAZIC, Valter**
  35576 CESSON SEVIGNE (FR)
- **SEIFI, Mozhdeh**
  35576 CESSON SEVIGNE (FR)
- **BURELLER, Olivier**
  35576 CESSON SEVIGNE (FR)

(74) Representative: **Huchet, Anne**
TECHNICOLOR
1-5, rue Jeanne d'Arc
92130 Issy-les-Moulineaux (FR)

(54) METHOD FOR DISPLAYING 4D LIGHT FIELD DATA

(57) In one embodiment, it is proposed a method for displaying/rendering a 4D light-field data. The method is executed by an electronic device, and is remarkable in that it comprises:
obtaining at least two Cartesian coordinates, each Cartesian coordinate being associated with a camera position in a three dimensional space;
obtaining at least one 4D light-field data associated with a selected Cartesian coordinates from said at least two Cartesian coordinates; and
rendering obtained 4D light-field data.

Tags			Data			
Parametrization plane offset z_2-z_1 (z_3 is relative to z_1) (float)			**Δz**			
Width in number of x_1 cells (integer)	min x_1 value (float)	max x_1 value (float)	**N**	$\mathbf{x_{1min}}$	$\mathbf{x_{1Max}}$	
Width in number of x_2 cells (integer)	min x_2 value (float)	max x_2 value (float)	**M**	$\mathbf{x_{2min}}$	$\mathbf{x_{2Max}}$	
Number of cameras (integer)			**n**			
Cam_1 focus (2 floats)			**x31**	**z31**		
lowest cam_1 intercept in (x_1,x_2)	cam_1 steepness	cam_1 number of digital lines in x_1,x_2	d_1	s_1	m(1)	
data line (m(1)*N * 3 bytes)			RGB_0	RGB_1	...	$RGB_{m(1)*N}$
...			...	...		
...			...	...	...	...
Cam_n focus (2 floats)			**x31**	**z31**		
lowest cam_n intercept in (x_1,x_2)	cam_n steepness	cam_n number of digital lines in x_1,x_2	d_n	s_n	m(n)	
data line (m(n)*N * 3 bytes)			RGB_0	RGB_1	...	$RGB_{m(n)*N}$

FIG.21

EP 3 145 192 A1

Description

Technical Field

[0001] The disclosure relates to the field of computational imaging, light-field acquisition devices, and plenoptic cameras. More precisely, it deals with a representation format for the Light-field that can be used for transmission, rendering, processing, mixing of light-field data.

Background

[0002] This section is intended to introduce the reader to various aspects of art, which may be related to various aspects of the present invention that are described and/or claimed below. This discussion is believed to be helpful in providing the reader with background information to facilitate a better understanding of the various aspects of the present invention. Accordingly, it should be understood that these statements are to be read in this light, and not as admissions of prior art.

[0003] The acquisition of 4D light-field data), which can be viewed as a sampling of a 4D light field (i.e. the recording of light rays as explained in Figure 1 of the article *:"* Understanding camera trade-offs through a Bayesian analysis of light field projections" by Anat Levin and al., published in the conference proceedings of ECCV 2008) is an hectic research subject.

[0004] Indeed, compared to classical 2D images obtained from a camera, 4D light-field data enable a user to have access to more post processing features that enhance the rendering of images and/or the interactivity with the user. For example, with 4D light-field data, it is possible to perform with ease refocusing of images a posteriori (i.e. refocusing with freely selected distances of focalization meaning that the position of a focal plane can be specified/selected a posteriori), as well as changing slightly the point of view in the scene of an image. In order to acquire 4D light-field data, several techniques can be used. For example, a plenoptic camera, as depicted in document WO 2013/180192 or in document GB 2488905**,** is able to acquire 4D light-field data. Details of the architecture of a plenoptic camera are provided in Figures 1 and 2 of the present document. Another way to acquire 4D light-field data is to use a camera array as depicted in Figure 3 of the present document. Details of an architecture of a camera array are provided in Figure 3. At last, another way of acquiring a 4D light field is to use a conventional camera that is configured to capture a sequence of 2D images of a same scene at different focal planes. For example, the technique described in the document "Light ray field capture using focal plane sweeping and its optical reconstruction using 3D displays" by J.-H. Park et al. , published in OPTICS EXPRESS, Vol. 22, No. 21 , in October 2014, can be used to achieve the acquisition of 4D light field data.

[0005] In the following only plenoptic camera and camera array are described, but it should be noted that the present disclosure can also be applied to other devices that acquire 4D light field data such as devices that comprise coded aperture elements as depicted in document US 2010/0265386**,** or in the article entitled "Image and depth from a conventional camera with a coded aperture" by A. Levin a al., published in the proceedings of SIGGRAPH 2007, or use wavefront coding techniques as mentioned in the article entitled "Extended depth of field through wave-front coding" by Edward R. Dowski, Jr., and W. Thomas Cathe, published in Applied Optics, 1995 Apr 10.

[0006] In the state of the art, there are several ways to represent (or define) 4D light-field data. Indeed, in the Chapter 3.3 of the Phd dissertation thesis entitled "Digital Light Field Photography" by Ren Ng, published in July 2006, three different ways to represent 4D light-field data are described. Firstly, 4D light-field data can be represented, when recorded by a plenoptic camera as the one depicted in Figure 1 for example, by a collection of micro-lens images (see the description of Figure 2 in the present document). 4D light-field data in this representation are named raw images (or raw 4D light-field data). Secondly, 4D light-field data can be represented, either when recorded by a plenoptic camera or by a camera array, by a set of sub-aperture images. A sub-aperture image corresponds to a captured image of a scene from a point of view, the point of view being slightly different between two sub-aperture images. These sub-aperture images give information about the parallax and depth of the imaged scene. Thirdly, 4D light-field data can be represented by a set of epipolar images (see for example the article entitled : "Generating EPI Representation of a 4D Light Fields with a Single Lens Focused Plenoptic Camera", by S. Wanner and al., published in the conference proceedings of ISVC 2011).

[0007] However, due to the fact that plenoptic devices are extremely heterogeneous (for example we can distinguish plenoptic devices of type 1 or 2 (which define particular optical arrangements of the micro-lenses (for example micro-lenses with different focal lengths, etc.))), and that camera arrays come in different flavors, it appears that all these light field acquisition devices have their proprietary file format, so that light-field technology cannot live besides regular 2D or 3D imaging as there is no standard supporting the acquisition and transmission of multi-dimensional information. For example, in the case that 4D light-field data is represented as raw 4D light-field data, some additional information such as the size of the micro-lenses, etc. are needed for the processing of the raw 4D light-field data, and for the moment

there isn't a standard that list all the needed additional information.

[0008] However, it should be noted that 4D light-field data as mentioned previously could also be represented by values derived from a parametrization process, as detailed in the paragraph III section A named "Light Field Representation" of the article entitled "Light Field Analysis for Modeling Image Formation" by Chi-Kai Liang and al., published in the IEEE Transactions on Image Processing, Vol. 20, N°2, in February 2011, or in the article entitled "Light Field Rendering" by Mar Levoy and Pat Hanrahan, or in the documents US 8,237,708**,** US 6,097,394 or US 6,023,523**.** Hence, 4D light-field data can be represented as a set of points (being associated with light rays) in a coordinate system (see paragraphs [0028] to [0030] of document WO 2013/180192). More precisely, the use of a parametrization of a 4D light-field via the use of two planes, either parallel or not, in order to represent 4D light-field data (i.e. each light ray is represented by a set of coordinates values corresponding to the intersection of such light ray with two planes) could be a way to avoid the use of proprietary formats, and to foster the interoperability between 4D light-field devices (i.e. plenoptic devices and/or camera arrays for example, but also for 4D light field display devices, such as the one described in the article entitled "Near-Eye Light Field Displays" by D. Lanman and D. Luebke, presented at the conference SIGGRAPH 2013 in July 2013).

[0009] Indeed, instead of sharing raw image files (which comprise minimally processed data from the image sensor of a light field camera) information with complex inter-leaving structure, it seems to be better to share a representation of the light rays that have been captured, with all their conventionality. Then the data can be processed, shared, transmitted and rendered independently of the way it has been acquired. It should be noted that a parametrized light field is also named a light slab representation in the previously mentioned article entitled "*Light Field Rendering*".

[0010] However, one drawback of this representation of a 4D light-field is that content to be stored is huge as mentioned in p.23 of the present document.

[0011] Indeed, using such representation implies to store for each parametrized ray the coordinates in the two planes (i.e. it is necessary to store a 4-uplet and the value of the radiance of such light ray). The process of converting a set of 2D images (defining a sampled light field) into a representation of light rays (each light ray being associated with a 4-uplet and a value of radiance) is depicted for example in document US 6,023,523

[0012] Hence, there is a need to provide a technique for providing a compact light field representation in the sense that it overcomes the previously mentioned storage issue.

Summary of the disclosure

[0013] References in the specification to "one embodiment", "an embodiment", "an example embodiment", indicate that the embodiment described may include a particular feature, structure, or characteristic, but every embodiment may not necessarily include the particular feature, structure, or characteristic. Moreover, such phrases are not necessarily referring to the same embodiment. Further, when a particular feature, structure, or characteristic is described in connection with an embodiment, it is submitted that it is within the knowledge of one skilled in the art to affect such feature, structure, or characteristic in connection with other embodiments whether or not explicitly described.

[0014] The present disclosure is directed to a method for rendering a 4D light-field data. Such method is executed by an electronic device, and is remarkable in that it comprises :

obtaining at least two Cartesian coordinates, each Cartesian coordinate being associated with a camera position in a three dimensional space;
obtaining at least one 4D light-field data associated with a selected Cartesian coordinates from said at least two Cartesian coordinates; and
rendering obtained 4D light-field data.

[0015] In a preferred embodiment, the method is remarkable in that obtaining said at least one 4D light-field data comprises receiving a digital lines representation of said at least one 4D light-field data.

[0016] In a preferred embodiment, the method is remarkable in that said obtaining at least one 4D light-field data comprises selecting data, based on said selected Cartesian coordinates, from a table.

[0017] In a preferred embodiment, said table is stored on distant server.

[0018] According to an exemplary implementation, the different steps of the method are implemented by a computer software program or programs, this software program comprising software instructions designed to be executed by a data processor of a relay module according to the disclosure and being designed to control the execution of the different steps of this method.

[0019] Consequently, an aspect of the disclosure also concerns a program liable to be executed by a computer or by a data processor, this program comprising instructions to command the execution of the steps of a method as mentioned here above.

[0020] This program can use any programming language whatsoever and be in the form of a source code, object code

or code that is intermediate between source code and object code, such as in a partially compiled form or in any other desirable form.

[0021] The disclosure also concerns an information medium readable by a data processor and comprising instructions of a program as mentioned here above.

[0022] The information medium can be any entity or device capable of storing the program. For example, the medium can comprise a storage means such as a ROM (which stands for "*Read Only Memory*"), for example a CD-ROM (which stands for "*Compact Disc - Read Only Memory*") or a microelectronic circuit ROM or again a magnetic recording means, for example a floppy disk or a hard disk drive.

[0023] Furthermore, the information medium may be a transmissible carrier such as an electrical or optical signal that can be conveyed through an electrical or optical cable, by radio or by other means. The program can be especially downloaded into an Internet-type network.

[0024] Alternately, the information medium can be an integrated circuit into which the program is incorporated, the circuit being adapted to executing or being used in the execution of the method in question.

[0025] According to one embodiment, an embodiment of the disclosure is implemented by means of software and/or hardware components. From this viewpoint, the term "module" can correspond in this document both to a software component and to a hardware component or to a set of hardware and software components.

[0026] A software component corresponds to one or more computer programs, one or more sub-programs of a program, or more generally to any element of a program or a software program capable of implementing a function or a set of functions according to what is described here below for the module concerned. One such software component is executed by a data processor of a physical entity (terminal, server, etc.) and is capable of accessing the hardware resources of this physical entity (memories, recording media, communications buses, input/output electronic boards, user interfaces, etc.).

[0027] Similarly, a hardware component corresponds to any element of a hardware unit capable of implementing a function or a set of functions according to what is described here below for the module concerned. It may be a programmable hardware component or a component with an integrated circuit for the execution of software, for example an integrated circuit, a smart card, a memory card, an electronic board for executing firmware etc. In a variant, the hardware component comprises a processor that is an integrated circuit such as a central processing unit, and/or a microprocessor, and/or an Application-specific integrated circuit (ASIC), and/or an Application-specific instruction-set processor (ASIP), and/or a graphics processing unit (GPU), and/or a physics processing unit (PPU), and/or a digital signal processor (DSP), and/or an image processor, and/or a coprocessor, and/or a floating-point unit, and/or a network processor, and/or an audio processor, and/or a multi-core processor. Moreover, the hardware component can also comprise a baseband processor (comprising for example memory units, and a firmware) and/or radio electronic circuits (that can comprise antennas) which receive or transmit radio signals. In one embodiment, the hardware component is compliant with one or more standards such as ISO/IEC 18092 / ECMA-340, ISO/IEC 21481 / ECMA-352, GSMA, StoLPaN, ETSI / SCP (Smart Card Platform), GlobalPlatform (i.e. a secure element). In a variant, the hardware component is a Radio-frequency identification (RFID) tag. In one embodiment, a hardware component comprises circuits that enable Bluetooth communications, and/or Wi-fi communications, and/or Zigbee communications, and/or USB communications and/or Firewire communications and/or NFC (for Near Field) communications.

[0028] It should also be noted that a step of obtaining an element/value in the present document can be viewed either as a step of reading such element/value in a memory unit of an electronic device or a step of receiving such element/value from another electronic device via communication means.

[0029] In another embodiment of the disclosure, it is proposed an electronic device for rendering 4D light-field data. Such electronic device is remarkable in that it comprises:

a first module configured to obtain at least two Cartesian coordinates, each Cartesian coordinate being associated with a camera position in a three dimensional space;
a second module configured to obtain at least one 4D light-field data associated with a selected Cartesian coordinates from said at least two Cartesian coordinates; and
a third module configured to render obtained 4D light-field data.

[0030] In a preferred embodiment, said second module is further configured to receive a digital lines representation of said at least one 4D light-field data.

[0031] In a preferred embodiment, said second module is further configured to select data, based on said selected Cartesian coordinates, from a table.

Brief description of the drawings

[0032] The above and other aspects of the invention will become more apparent by the following detailed description

of exemplary embodiments thereof with reference to the attached drawings in which:

- Figure 1 is a schematic diagram of a plenoptic camera being a light field camera according to an embodiment of the disclosure;
- Figure 2 present another view of the sensor array disclosed in Figure 1;
- Figure 3 is a schematic diagram of a multi-camera array being a light field camera according to an embodiment of the disclosure;
- Figure 4 presents the parametrization of a light ray by the use of at least two planes according to an embodiment of the disclosure;
- Figures 5(a) to (h) show two different representations of 2D slices according to an embodiment of the disclosure;
- Figure 6 presents a 2D ray diagram of a random ray fan according to an embodiment of the disclosure;
- Figures 7(a)-(b) present the result of light field filtering from the 2D ray diagram of Figure 6, according to an embodiment of the disclosure;
- Figure 8 presents a coordinate system used for general viewpoint rendering according to an embodiment of the disclosure;
- Figure 9 depicts a selection of rays according to an embodiment of the disclosure;
- Figure 10 presents a rendering ray trace of selected rays of Figure 9, according to an embodiment of the disclosure;
- Figure 11 presents a 2D ray diagram with selected rays according to an embodiment of the disclosure;
- Figure 12 shows the rendering ray trace of selected rays of Figure 11, according to an embodiment of the disclosure;
- Figure 13 presents a 2D ray diagram with selected rays according to an embodiment of the disclosure;
- Figure 14 shows the rendering ray trace of selected rays of Figure 13, according to an embodiment of the disclosure;
- Figure 15 is a 2D ray diagram of two groups of three cameras according to an embodiment of the disclosure. The first group has the pinholes set at the plane $z_3 = 2$, and the second set at the plane $z_3 = 14$;
- Figure 16 is a sonogram of the Radon transform of the light field representation of Figure 15;
- Figure 17 presents a coordinate system used for general viewpoint rendering in 3D space according to an embodiment of the disclosure;
- Figure 18 is a 2D ray diagram of rays captured by one camera at coordinates $x_3 = 2$, $z_3 = 2$ with an aperture of $|A| < 0,5$, according to one embodiment of the disclosure. The lines used to parametrize are sampled by 256 cells;
- Figure 19 presents an example of a line in a 2D ray diagram according to an embodiment of the disclosure;
- Figure 20 is a 2D ray diagram comprising six lines or digital lines according to an embodiment of the disclosure;
- Figure 21 is an example of a representation of light field rays according to an embodiment of the disclosure;
- Figure 22(a) is another example of a representation of light field rays according to an embodiment of the disclosure;
- Figure 22(b) is another example of a representation of light field rays according to an embodiment of the disclosure;
- Figure 22(c)-(d) illustrate an enclosed object into a double cube according to an embodiment of the disclosure;
- Figure 23 presents the result of the application of the Discrete Radon Transform on data from Figure 18;
- Figure 24 is a zoom on an area in Figure 23;
- Figure 25 presents on the left hand side a 2D ray diagram in x and the result of the application of the Discrete Radon Transform on data in such 2D ray diagram, and on the right hand side, a 2D ray diagram in y and the result of the application of the Discrete Radon Transform on data in such 2D ray diagram;
- Figure 26(a) presents some steps of a method for providing a compact light field representation according to an embodiment of the disclosure;
- Figure 26(b) presents a synthetic scheme that depicts a conversion from either 4D light-field data or a conventional 2D image into a representation based on the parametrization of light rays via the use of two planes according to an embodiment of the disclosure;
- Figures 26 (c)-(d) present how a ray is "extracted" from a pixel (or virtual pixel) sensor, according to an embodiment of the disclosure;
- Figure 27 presents a menu that comprises a 3D representation from which a user can select a position of a camera associated with 4D light-field data, according to an embodiment of the disclosure;
- Figure 28 presents an example of an electronic device that can be used to perform one or several steps of methods disclosed in the present document.

Detailed description

[0033] Figure 1 present schematically the main components comprised in a plenoptic camera. More precisely, a plenoptic camera comprises a main lens referenced 101, and a sensor array (i.e. an array of pixels)., referenced 104. Between the main lens 101 and the sensor array 104, a microlens array referenced 102, that comprises a set of micro lenses referenced 103, is positioned. It should be noted that optionally some spacers might be located between the micro-lens array around each lens and the sensor to prevent light from one lens to overlap with the light of other lenses

at the sensor side. Hence, a plenoptic camera can be viewed as a conventional camera plus a micro-lens array set just in front of the sensor as illustrated in Figure 1. The light rays passing through a micro-lens cover a part of the sensor array that records the radiance of these light rays. The recording by this part of the sensor defines a micro-lens image.

[0034] Figure 2 present another view of the sensor array 104. Indeed, in such view, it appears that the sensor array 104 comprises a set of pixels, referenced 201. The light rays passing through a micro-lens cover a number of pixels 201, and these pixels record the energy value of light rays that are incident/received.

[0035] Hence the sensor array 104 of a plenoptic camera records an image which comprises a collection of 2D small images (i.e. the micro-lens images referenced 202) arranged within a 2D image (which is also named a raw 4D light-field image). Indeed, each small image (i.e. the micro-lens images) is produced by a lens (the lens can be identified by coordinates (i,j) from the array of lens). Hence, the pixels of the light-field are associated with 4 coordinates (x,y,i,j). L(x,y,i,j) being the 4D light-field recorded by the sensor illustrates the image which is recorded by the sensor. Each micro-lens produces a micro-image represented by a circle (the shape of the small image depends on the shape of the micro-lenses which is typically circular). Pixel coordinates (in the sensor array) are labelled (x,y). p is the distance between 2 consecutive micro-images, p is not necessarily an integer value. Micro-lenses are chosen such that p is larger than a pixel size δ. Micro-lens images are referenced by their coordinate (i, j). Each micro-lens image samples the pupil of the main-lens with the (u,v) coordinate system. Some pixels might not receive any photons from any micro-lens especially if the shape of the micro-lenses is circular. In this case, the inter micro-lens space is masked out to prevent photons to pass outside from a micro-lens, resulting in some dark areas in the micro-images. (if the micro-lenses have a square shape, no masking is needed). The center of a micro-lens image (i,j) is located on the sensor at the coordinate $(x_{i,j},y_{i,j})$. θ is the angle between the square lattice of pixel and the square lattice of micro-lenses, in Figure 2 $\theta = 0$. Assuming the micro-lenses are arranged according to a regular square lattice, the $(x_{i,j}, y_{i,j})$ can be computed by the following equation considering $(x_{0,0}, y_{0,0})$ the pixel coordinate of the micro-lens image (0,0):

$$\begin{bmatrix} x_{i,j} \\ y_{i,j} \end{bmatrix} = p \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} + \begin{bmatrix} x_{0,0} \\ y_{0,0} \end{bmatrix}$$

Figure 2 also illustrates that an object from the scene is visible on several contiguous micro-lens images (dark dots). The distance between 2 consecutive views of an object is w, this distance is named the replication distance. Hence, an object is visible on r consecutive micro-lens images with :

$$r = \left\lfloor \frac{p}{|p - w|} \right\rfloor$$

r is the number of consecutive micro-lens images in one dimension. An object is visible in r^2 micro-lens images. Depending on the shape of the micro-lens image, some of the r^2 views of the object might be invisible.

[0036] As mentioned previously in this document, a 4D light-field image can be represented by sub-aperture images. Each sub-aperture image is composed of the pixels of same position selected from each microlens image. Such representation is possible for plenoptic cameras of type 1.

[0037] According to the location of the pixel, multiview sub-aperture images can be obtained and have different information of incident rays respectively. The conventional photograph is equal to the integration of all sub-aperture images, summing all the incident light.

[0038] Hence, a 4D light-field image represented by sub-aperture images is defined as being a set (or collection) of $m \times n$ sub-aperture images where each sub-aperture image corresponds to a slightly different perspective of a same scene, and where the parameters *m* and *n* are integer greater or equal to one. Such 4D light-field image is also named a matrix of view in the state of the art. In order to obtain such representation, the pixels from the raw sensor image are re-arranged to create an array of images, with the pixels in each coming from the same position under every micro-lens. Each sub-aperture image can be thought of as capturing a traditional image of the light passing through only a small sub-aperture region of the camera's lens. While these images are quite similar, they each have a unique optical center within the camera's aperture plane, so the same real-world object will appear in different locations of each sub-aperture image. That is, the sub-aperture images provide parallax on the captured scene: foreground objects have relatively higher displacements between adjacent sub-aperture images than background objects.

[0039] Usually, in the case that a light field image is acquired by a plenoptic camera (of type 1.0), the number of pixels positioned below a micro-lens determines the number of sub-aperture images, and the number of micro-lens determines the number of pixels in each sub-aperture images.

[0040] More details related to plenoptic camera can be found out in the Section 4 entitled "*Image formation of a Light field camera*" in the Article "The Light Field Camera: Extended Depth of Field, Aliasing, and Superresolution" by Tom

E. Bishop and Paolo Favaro, published in the IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34, N°5, in May 2012.

[0041] It should be noted that the present technique can also be applied on "conventional camera" (in the sense that no additional micro-lens array is positioned between the main lens and array of pixels), in the case that at least a part of the pixels of such conventional camera are designed in the same way (or similar way) as the one described in the document US2013258098**.** Indeed, document US2013258098 discloses a pixel that can record light field data due to the use of several light receiving sections (for example referenced 116 and 117 in document US2013258098). Hence, one skilled in the art could assimilate such conventional camera with an array of pixels integrating the technique of document US2013258098 as a kind of plenoptic camera as depicted in Figure 1, in which each micro-lens concentrates light rays on two pixels comprised in the sensor 104. It should be noted that technique of document US2013258098 can be generalized in the sense that a pixel can record more than two data information (obtained by the two low and high receiving sections), if more receiving section are integrated in the architecture of a pixel. The present disclosure can reduce the aberrations of the main lens of such "conventional camera" integrating pixels that can record light field data as mentioned previously.

[0042] It should also be noted that the present disclosure can also be applied to other devices that acquire 4D light field data such as devices that comprise coded aperture elements as depicted in document US 2010/0265386**,** or in the article entitled "Image and depth from a conventional camera with a coded aperture" by A. Levin a al., published in the proceedings of SIGGRAPH 2007, or use wavefront coding techniques as mentioned in the article entitled "Extended depth of field through wave-front coding" by Edward R. Dowski, Jr., and W. Thomas Cathe, published in Applied Optics, 1995 Apr 10.

[0043] Figure 3 presents in a schematic way a multi-camera array referenced 300. The multi-camera array 300 comprises an array of lenses, referenced 301, comprising several lenses referenced 302, and one or several sensor arrays, referenced 304. The multi-camera array 300 is without main lens. The array of lenses is often a small device, which is commonly named a micro-lens array. It is worth noting that the multi-camera array with a single sensor can be considered as a special case of plenoptic camera where the main lens has an infinite focal. Indeed, a lens with an infinite focal distance has no impact on the rays of light.

[0044] For example, a multi-camera array can be a Pelican array camera as the one depicted in the document WO 2014149403 A1**,** or a multi camera array as the one developed by Standford University in the article entitled "High Performance Imaging Using Large Camera Arrays", by B. Wilburn et al., published ACM Transactions on Graphics, Vol 24, No 3, July 2005, pp. 765-776 (Proceedings of ACM SIGGRAPH 2005)..

[0045] Figure 4 presents how to parametrize a light ray, referenced 401, by the use of two planes, referenced 402 and 403, in 3D space. Each plane 402, 403is defined by two normal vectors (for plane 402, the vectors $(\vec{i}_1,\vec{j}_1)$ defines it, and a 2D coordinates system can be used (i.e. the coordinates system defined by the orthogonal axes (x_1) and (y_1), and for plane 403, the vectors $(\vec{i}_2,\vec{j}_2)$ defines it and a 2D coordinates system can be used (i.e. the coordinates system defined by the orthogonal axes (x_2) and (y_2)). When a light ray intersects a plane, it is possible to determine in 3D space the coordinates of the intersection point (that comprise three components). However, in order to obtain the direction of the light ray it is necessary to use another plane for identifying the direction of the light ray in the 3D space. By considering that the two planes used for the parametrization are parallel and have known z values (in the 3D Cartesian coordinate space), it is possible to identify a light ray by only four coordinates (two values in the 2D coordinates system defined by the orthogonal axes (x_1) and (y_1), and two values in the 2D coordinates system defined by the orthogonal axes (x_2) and (y_2)), and a value L which corresponds to the intensity/radiance of such light ray (see more details on such parametrization in the previously mentioned article entitled "*Light Field Rendering*")). Hence, a light field, which can be viewed as a collection of light rays in a scene, can be parametrized by a pair of planes, and each light ray of the light field is represented as a point with four coordinates (for example the light ray 401, which has coordinates $(a,b,c,d) \in \mathbb{R}^4$, and a radiance value $L(a,b,c,d) \in \mathbb{R}$. By abuse of notation, the coordinate of a light ray are noted $(x_1,y_1,x_2,y_2) \in \mathbb{R}^4$ which could be misleading as the terms x_1,y_1,x_2,y_2 also mentioned axis).

[0046] It should be noted that a light ray can also be projected (via an orthographic or orthogonal projection)onto either a plane referenced 404 or a plane referenced 405. The plane 404 is a plane which is perpendicular to the two planes 402 and 403, in which y in the 3D Cartesian coordinate space has a constant value. The plane 405 is a plane which is perpendicular to the two planes 402 and 403, in which x in the 3D Cartesian coordinate space has a constant value. The result of the projection of some light rays onto plane 404 are depicted in Figures 5(a), 5(c), 5(e), and 5(g). Then, in Figures 5(b), 5(d), 5(f) and 5(h) a representation in a 2D ray diagram of the rays depicted respectively in Figures 5(a), 5(c), 5(e), and 5(g) is given. Indeed, the projected light ray (as represented in Figures 5(a), 5(c), 5(e), and 5(g)) is

represented by a point in a 2D ray diagram (as also explained in Figure 4B of document WO 2013/180192). It should be noted that similar Figures can be obtained by considering projections on the plane 405.

[0047] Indeed, as the light-field is a function of four variables, it is very difficult to illustrate in general. However, we can take a 2D slice and figure out in two different ways how particular light ray sets map into 2D graphs.

[0048] Figures 5(a) to (h) show two different representations of such 2D slices restricted to the x_1 and x_2 axis. Figure 5(a) shows a collimated ray set, Figure 5(c) presents a divergent ray fan, Figure 5(e) presents a convergent ray fan, and Figure 5(g) presents a convergent ray fan, focused on the plane 403..Figures 5(b), 5(d), 5(f) and 5(h) present another way of plotting light-field data (i.e. ray lights with their direction) into a 2D ray coordinate system which is maybe easier to handle or create but more difficult to visualize.

[0049] Beginning with Figure 5(b), where we have plotted in the 2D ray diagram, collimated raysets with various angle of incidence. When the ray set is parallel to the z axis, then each light ray is plotted along a 45° line (due to the fact that $x_2 = x_1$ + constant). When the rayset points upward, the line shifts upward in the 2D diagram, and when the rayset points downward, the line shifts also downward (negative x_2). The conclusion is that parallel rays will map onto a line at 45°, and if the parallel rayset makes an angle with the z axis, the x_2 line intercept shifts accordingly.

[0050] Figure 5(d) presents a diverging beam. Indeed, various degrees of divergence are plotted. The markers on the red line represent a set of rays diverging from infinity. Basically, this means that the set is collimated. The Figure 5(a) showed already that the line is at 45° and that the relationship between x_1 and x_2 is linear. As the point of divergence is moved from infinity toward the axis x_1, the 2D ray diagrams is still linear but the steepness of the lines increases with the amount of divergence.

[0051] At the limit, if the point, from where the set of rays diverges, is on the x_1 axis, the rays would be plotted along the vertical axis x_2 on the 2D ray diagram.

[0052] Figure 5(e) presents a convergent ray set. Again, the red marker line is at 45° and represents a beam converging from infinity, which is again a parallel set of rays. It should be noticed that when the point of convergence is pulled toward the x_2 axis, the rays would map on lines of decreasing slopes.

[0053] Figure 5(g) is the limit when the convergence point is on x_2, a focused rayset, then all rays on the diagram are one the horizontal axis.

[0054] It should be noted that in the case that two 2D ray diagrams are used (one derived from the projection onto the plane 404 (see Figures 5(a), 5(c), 5(e), and 5(g)), noted $\Pi(x_1,x_2)$, and one obtained from the projection onto the plane 405, noted $\Pi(y_1,y_2)$) for representing the light rays, it is necessary to store another information that enables the linking between a point in one 2D ray diagram (i.e. a point in $\Pi(x_1,x_2)$) with a point comprises into the other2D ray diagram (i.e. in $\Pi(y_1,y_2)$), or vice-versa. In the Figure 26, a technique that enables to bypass the storing of such explicit additional information is described for all the light rays in a sampled light field. However, in order to understand such technique, some remarks and comments should be given, in view of Figures 6 to 24.

[0055] Figure 6 presents a 2D ray diagram of a random ray fan.

[0056] In order to generate a random light-field, a uniform random number generator can be used. Then a 2D ray diagram as the basic representation of the light-field and then regenerate in a simple way chosen portions of the light-field from the format.

[0057] The x value will extend over [-10; 10], the direction of propagation is z, the second plane for parametrization is placed at z = 1 and the rays will be assigned with a random direction. In order to generate uniformly distributed random ray directions, it is better to randomly generate the x_2 component with a uniform number generator, for instance inside of [-15; 15]. Figure 6 shows the corresponding 2D ray-diagram. As expected, there are rays everywhere within $\|x_1\| < 10$ and $\|x_2\| < 15$, where $\|.\|$ is the Euclidian norm.

[0058] We would like here to get proofs for observations previously identified in Figures 5(a)-(g). It will give us some formula for particular renderings. The questions that need to be answered among others are: how to render a viewpoint, how to refocus elsewhere, what rays in the 2D ray diagram are used for what purpose? There could be millions of points in a 2D ray-diagram like the one of Figure 6. Do we need to test or trace all of them to render some particular image?

[0059] In computer graphics, renderings use pinhole cameras to generate images. An image is formed on a plane by filtering the light-field by a tiny hole some focal distance in front of that plane (see for example the previously mentioned article entitled:" *Understanding camera trade-offs through a Bayesian analysis of light field projections*"). Hence, rendering images is a matter of "holes". From a ray stand point, this is equivalent to search for convergence or divergence of a set of rays.

[0060] Figure 17 presents the notation used to analyze the relationship between the light-field and the ray diagram. From this figure, if $z_1 \neq z_2$, and $z_3 \neq z_1$:

$$\frac{x_1 - x_3}{z_1 - z_3} = \frac{x_2 - x_1}{z_2 - z_1} \Leftrightarrow L_{2D}(x_1, x_2) = (z_2 - z_3)x_1 + (z_3 - z_1)\, x_2 = (z_1 - z_3)x_3$$

[0061] This formula, referenced as **equation 1**, links the pinhole, or the viewpoint (x_3,z_3) to the recorded lightfield parameters (x_1,z_1,x_2,z_2) which is the position in the 2D ray-diagram since z_1 and z_2 are known. Equation 1 is, as expected from the previous section, a line of the form ax + by = c. Also, as expected, its slope (the coefficient of x_1 is a function of the depth of the rendering pinhole at z_3 to one of the plane and the y intercept is a function of the off-axis position x_3. of the pinhole. Let us now consider some limit cases of that equation:

a) On axis collimated ray fan ($z_3 \to \pm\infty$, $x_3 = 0$)

[0062] If $z_3 \to \pm\infty$ and $x_3 = 0$, we are in the presence of a collimated ray fan parallel to the z axis and from equation 1,

$$\lim_{z_3 \to \pm\infty} L_{2D}(x_1, x_2) = x_2 - x_1 = 0$$

which is indeed a linear relationship $x_2 = x_1$. Hence, if from the 2D ray-diagram of Figure 6, we only select and render rays that obey the relationship $x_2 = x_1$, we a going to trace only the rays from the light-field that propagate parallel to the z axis.

b) Off axis collimated ray fan ($z_3 \to \pm\infty$, $x_3 = \neq 0$)

[0063] For off-axis collimated ray fans, there is an affine relationship between z_3 and x_3: $x_3 = az_3 + b$so that if we put that into equation 1:

$$\lim_{z_3 \to \pm\infty} L_{2D}(x_1, x_2) = x_2 - x_1 = a(z_2 - z_1) \Leftrightarrow x_2 = x_1 + a(z_2 - z_1) \quad ,$$

which is illustrated in Figure 5(a) where off axis beams are 45° lines with y intercepts varying as the slope the collimated beams makes with the z axis. Again, if from the 2D ray-diagram, we only select and render light rays that obey the relationship $x_2 = x_1 + c$, we are going to trace light rays from the whole light field that propagate as a parallel ray set with some angle with respect to the z axis. The Figures 7(a) and 7(b) illustrate exactly this case, by selecting the rays from the 2D diagram of Figure 6, we see that off-axis parallel rays are traced. Indeed, Figure 7(a) and 7(b) presents Light field filtering and reconstruction of light rays from the 2D ray diagram of Figure 6. More precisely, the Figure 7(a) depicts the selection of the light rays in the 2D ray diagram of Figure 6, according to a selection criteria being $\|L_{2D}(x_1,x_2) - (x_1 + 5)\| < 0.1$, which enables the selection of parallel set of rays with a direction propagation as shown in Figure 7(b). Indeed, the Figure 7(b) depicts the reconstructed light field from the selected rays in the 2D ray diagram of Figure 6.

c) Focused ray fan

[0064] By "focused" we mean a fan that converges to a point after the second sampling line at $z = z_2$. In this case, we have $z_3 > z_2$ and $z_2 > z_1$, then equation 1 can be written as

$$x_2 = \frac{z_3 - z_2}{z_3 - z_1} x_1 + \frac{z_2 - z_1}{z_3 - z_1} x_3$$

[0065] And the conditions $z_3 > z_2$ and $z_2 > z_1$, imply that

$$0 < \alpha = \frac{z_3 - z_2}{z_3 - z_1} < 1 \text{ and } 0 < \beta = \frac{z_2 - z_1}{z_3 - z_1} < 1.$$

[0066] On axis ray fan converging to $x_3 = 0$, like on Figures 5(e) and 5(f), the relationship describes lines with a slope angle smaller than 45° ($\alpha < 1$).

[0067] Let us demonstrate how this is practically used. Let's assume that we have two planes sampled light field from which we build up a ray diagram. We are still in a 2D slice of the 4D light field. This ray diagram is the one from Figure 6. Let's assume also that we would like to render three views. For that purpose, we define three pinholes, or three points after $z = z_2$, where rays shall converge to. We know now that those rays can be selected on the 2D ray-diagram along

line $x_2 = \alpha x_1 + \beta x_3$, where $|a| < 1$.

[0068] We define three cameras located respectively at position C_0 ($z_3 = +2, x_3 = 0$), C_{-1} ($z_3 = +2, x_3 = -2$) and C_{+1} ($z_3 = +2, x_3 = +3$). We set $z_1 = 0$ and $z_2 = 1$. Once those values are set in equation 1, we have to select the rays on the 2D ray-diagram that obey the three relationships:

$$x_2 = \frac{x_1}{2}$$

$$x_2 = \frac{x_1 - 2}{2}$$

$$x_2 = \frac{x_1 + 3}{2}$$

[0069] The Figure 9 presents a selection of the light rays in Figure 6 with a selection criteria being $\left|\left|L_{2D}(x_1, x_2) - {}^{x_1}/_{2}\right|\right| \leq 0.1.$ The Figure 10 depicts the rendering of the selected light rays (as selected in Figure 9).

d) Divergent ray fan

[0070] With "divergent", it is meant a ray fan that appears to diverge from a point ($z_3 < z_1, x_3$) set before the first line at $z = z_1$. With $z_3 < z_1 < z_2$, we get that $z_2 - z_3 > z_1 - z_3 > 0$, and the slope α is then:

$$\alpha = \frac{z_3 - z_2}{z_3 - z_1} > 1.$$

[0071] Divergent beams map into the 2D ray diagram as lines with a slope greater than 45°.

e) Divergent ray fan on the first parametrization line

[0072] The point of divergence can be placed at $z_3 = z_1$. In this case, equation 1 can be rewritten, and we get $x_1 = x_3$. To ray trace those rays that appear to diverge from the first plane, we have to select vertical lines on the 2D ray diagram.

f) Convergent ray fan on the second parametrization line

[0073] From equation 1, if we set $z_3 = z_2$, then we get $x_2 = x_3$. Ray fan converging to the second plane map as horizontal lines on the 2D ray diagram. The Figures 11 and 12 have been obtained with (z_3, x_3) = (1,5).

g) The case of real lenses

[0074] Pinholes are abstract objects, if we want to render the light field by using real lenses models, we need to know the position and the size of the entrance pupil of the lens, and then select in the ray diagram those rays that will trace through the entrance pupil. If the entrance pupil is at (z_l, x_l), and if it has an aperture diameter A with the lens axis parallel to z, then we need to select in the ray diagram all ray points of the ray diagram whose coordinates satisfy equation 1, but with x_3 varying inside $\left[x_l - \frac{A}{2}, x_l + \frac{A}{2}\right]$.

[0075] In the Figures 13 and 14, the entrance pupil is at (z_l, x_l) = (4,7), its size A = 6, and due to this size, there is a whole set of parallel lines in the 2D ray diagram that need to be selected and trace the rays belonging to the whole line set.

[0076] Now, we focus on observations of the application of the Radon Transform to a 2D ray diagram. As mentioned previously, if there are pinholes toward which rays do concentrate, the two plane parametrization of those rays are represented as lines in the 2D ray diagram. Let us for the sake of illustration set up a light-field that is captured by two

groups of three cameras. The first group of cameras is at a first depth(x_3,z_3) = {(10,2); (2,2); (-5,2)}, and the second group at another depth (x_3,z_3) = {(6,14); (-3,14); (-9,14)}. All cameras have entrance pupil of 0.1 units (units can be either meters, centimeters, etc.). The parametrization planes are set at $z_1 = 0$; $z_2 = 1$, and the size of the planes are $|x_1| \leq 10$; $|x_2| \leq 10$ units. Both planes are sampled by 1000 cells, so that the 2D ray diagram is represented as a 1000 × 1000 pixel image (see Figure 15).

[0077] Then as it has been taught previously, just by observing Figure 15(which is a 2D ray diagram of two groups of three cameras, the first group having the pinholes set at the plane $z_3 = 2$, and the second set at the plane $z_3 = 14$), we see that there are two set of three lines, each group is set at a different depth, because on the 2D ray diagram, different depths result in different slopes. Hence from this observation, we see that in this diagram, only certain lines are important for the transmission or for the rendering of the light-field, so that there is room for compressing efficiently this representation.

[0078] A natural way of defining which values from the 2D ray-diagram are important is to detect all the lines from the diagram. This can be achieved via the use of Radon transform. The Radon transform in two dimensions, named after the Austrian mathematician Johann Radon, is defined as the integral transform consisting of the integral of a function over straight lines.

[0079] This transform maps the 2D ray diagram into a parameter space which efficiently characterize a light field. Figure 16 is a sinogram, which is the visualization of the Radon transform applied on the2D ray diagram of Figure 15. We see that it exhibits six local maxima in two groups of three. The first one along the line $\theta \approx 42$ and the second along $\theta \approx 27$ degrees.

[0080] Figure 17 presents a coordinate system used for general viewpoint rendering in a 3D space.

[0081] Due to the parametrization described in Figure 4, it is also possible to demonstrate that the following relationship stands between a point where we would like to render the parametrized light field at (such point being positioned in a plane located $z = z_3$, see Figure 17) and the intersections points with the two planes used in the parametrization process (see the two planes perpendicular to z axis respectively at $z = z_1$ and $z = z_2$ in Figure 17):

$$(z_2 - z_3)(x_1 + y_1) + (z_3 - z_1)(x_2 + y_2) = (z_2 - z_1)(x_3 + y_3)$$

[0082] As all z_i are known, the only parameters of this equation, referenced as equation 2,are the four parameters (x_1,y_1,x_2,y_2). Equation 2 basically means that all the rays that do impact (or converge to, or an equivalent formulation would be "form an image") through the point (x_3,y_3,z_3) have their 4 coordinates linked.

It should also be noted that equation 2 defines a hyperplane in $\mathbb{R}^4$ (see the article entitled "Polyhedral geometry and the two-plane parameterization", by X. Gu and al., published in Rendering Techniques'97, Proceedings of the Eurographics Workshop in St. Etienne, France, June 16-18, 1997.

[0083] Again, this means that if we want to render images from a two-plane parametrized light-field, we will only render the rays in the vicinity of hyperplanes. There is no need to trace them all. Of course everything behaves well, since we always chose parametrization planes in the direction of propagation or direction of rendering.

Some comments about a representation format in 2D

[0084] If the light-field is parametrized by two planes, its representation in the phase space (i.e. a 4D ray diagram) occupies a portion of a 4D space. As the planes have finite dimensions, the 4D support of the plenoptic function in that phase space is compact. We can estimate the space required to store a light-field parametrized in this way. Let us consider both the cases of a 2D slice and the whole 4D case. Let us also assume that the planes are sampled by 1000 × 1000 cells. This means that the 2D phase space is represented by a matrix of 1000 × 1000 cells, and the 4D phase space by a matrix of 1000^4 cells. If the spectrum of the ray associated to a cell is sampled by three bands, typically red (R), green (G) and blue (B) quantized with 8 bits, the color is stored as a 24 bits or 3 bytes value. We can see that each 4D representation in this particular case requires 2.8 TBytes of storage.

[0085] From Figure 15, we see also that the matrix will be very sparsely populated, there will be lots of cells without any value at all, and this case is general with the light-field, since it is going to be captured by discrete acquisition means. The 4D representation matrix will even be less populated proportionally, so that it does not make sense to manipulate huge matrices. Similarly to what is done in sparse matrix representation, maybe we can find a way to avoid it.

[0086] We have also seen that for the 2D sliced representation, the rays do map along the vicinity of lines. In the 4D case, they map in the vicinity of hyper-planes.

[0087] Let us restrict the following discussion to the 2D case. As we have a mean to find the parameters of the lines with the discrete Radon transform, we can locate the regions of the matrix where there are representative values.

[0088] In order to illustrate explanations, we will take an example to follow the main ideas exposed hereafter. Figure 18 shows 2D ray diagram of a system as described in the caption. More precisely, Figure 18 depicts a 2D ray diagram of rays captured by one camera at position $x_3 = 2$, $z_3 = 2$, with an aperture of $|A| < 0.5$, and where the lines used to parametrized are sampled by 256 cells. If the image shown on Figure 18 is interpreted as a matrix, we can see that is very sparsely populated. The question that arises is: how is it possible to extract only the representative data from that matrix? What is the best way to store it in a file (In a structured manner, of course)? Even if the rays are saved individually in a file instead of the 4D phase space matrix, this requires to save for each ray, at least 2 bytes for each position x_i or y_i plus 3 bytes for the color. This is 7 bytes per ray for a 2D sliced light-field, and 11 bytes per ray for its full 4D representation. And even then, the rays are stored randomly in the file which might be very improper for a lot of applications that need to manipulate the representation.

A beam of digital lines

[0089] As we know that the rays are mapped along lines, it is more efficient to store in sequence the parameters of the line (slope related s and intercept d) and then the collection of rays belonging to that line, and then the next line parameters and respective rays, and so on.

[0090] This would require 2 bytes for s and 2 for d and then only 3 bytes per ray, which, if we ignore the overhead of line parameters in comparison to color data, offers a compression ratio of 1:2 and 1:4 for the 2D, respectively 4D ray representation. Moreover, the rays would be ordered along lines in the file. In order to set lines through matrix cells we need to define the so called digital lines which approximate them with minimum error. We follow common definitions and digital lines defined in literature dedicated to the discrete Radon transform (see for example the article entitled "A fast digital radon transform-an efficient means for devaluating the hough transform" by W. A. Götz and H. J. Druckmüller, published in Pattern Recogn., 28(12):1985-1992, Dec. 1995, and in the article entitled "Discrete Radon transform has an exact, fast inverse and generalizes to operations other than sums along lines.", by W. H. Press, in Proc. Natl. Acad. Sci. USA, 103(51):19249-19254, 2006.).

[0091] At this stage the digital lines are generated with Bresenham's algorithm. More advanced recursive type digital line generation will be shown later on. From previous references, the digital line is defined by two points of the grid, (0, d) and (N - 1, s) where the parameter s is coupled with the slope of a line by the following equation: $a = \frac{s}{N-1}$ where $s \in \{0,1, ...,N - 1\}$. d is the intercept and $d \in \{0,1, ..., N - 1\}$

[0092] See for example the Figure 19, which is an illustration of d, s and numbering of the x axis for the digital lines. As we have seen it previously, we can restrict the digital line to one octant, either the first one within $\left[0, \frac{\Pi}{4}\right]$, or the second octant $\left[\frac{\Pi}{4}, \frac{\Pi}{2}\right]$.

[0093] Until now, we have only two points of a line. This is sufficient for an analytical line, but now we also have to provide the grid points that approximate the analytical line. The most straightforward way to get a digital line is to round an analytical line to its nearest grid-point. Bresenham's algorithm provides a way to do it with minimal operation number whereas the approach developed in the previously mentioned articles, which gives alternate definition which are better adapted to a fast discrete Radon transform calculation. The Bresenham implementation has been adapted from the following reference: http://www.cs.helsinki.fi/group/goa/mallinnus/lines/bresenh.html. As it has been previously said, the analysis is restricted to one octant. In this particular case, we have $0 \leq a \leq 1$ so that $d \leq s$.

[0094] Consider now a family of digital lines having the same slope a but different intercepts d, and the family of lines being contiguous. This family of digital lines will be named a "beam". Figure 20 shows a beam of 6 digital lines according to that definition, with the same $s - d$ value, but different d intercepts. A particular pixel within the region hit by the beam of lines belongs only to one unique line.

[0095] Hence, to sum up, ray data parameterized by a sampled pair of lines (in 2D) and belonging to one camera, belong to a family of digital lines (beam) in the phase space used for representing the data. The header of the beam can simply contain the slope a and the thickness $d_{max} - d_{min}$ of the beam. The ray values will be stored as RGB colors along digital lines whose header can be d and s. The file will not store void cells of the ray diagram sampled space. Nor will it store the coordinates x_1, x_2 of the rays, the later will be deduced from d, s and from the position of the cell along the digital line. The Figure 21 is a format description that takes into account these remarks, as described later.

Discrete Radon transform for characterizing a beam of digital lines

[0096] In order for the previous format to be implementable and usable as an interchange and rendering basis, it is

at least wishful to be able to measure or guess the basic parameters which have to be written inside of the format. The parameters that need to be estimated from the lightfield or from camera's geometry are the slope *a* the lower and upper bounds of the digital line intercepts (d_{min},d_{max}), and then the digital line parameters (d_i,s_i). The discrete Radon transform has been discussed as a tool to measure the support location of the light-field in the ray diagram.

[0097] The Figure 23 shows the discrete Radon transform in the digital line parameter space (*d*, *s*), which shall not be confused with (ρ,θ) space of the sinogram.

[0098] More precisely, Figure 23 shows the discrete Radon transform of the distribution from Figures 18, and Figure 24 is a zoom into the region of interest comprised in Figure 23. The beam of digital lines is spotted by the search for the maximum value parameters. There could be some offset between the geometrical center of symmetry of the DRT and the actual position of the maximum due to image content so that later on, an algorithm has to be found to pin-point the center of symmetry instead of the maximum. Instead of summing up the RGB values along the integration lines, finding the maximum without the bias of the content could be done by summing up the number of rays per cell along the integration line, which is more a Hough transform (binary version of Radon). Then, the waist of the beam transform as shown on Figure 24 is easy to find and it directly gives the values (d_{min}, d_{max}) used into the format. In fact the point (d_{min} = 74, *s* = 201) is the lower envelope of the beam of digital lines from Figure 18, and (d_{max} = 81, *s* = 208) is the upper envelope of the beam of digital lines.

[0099] Once we have got those values, we can begin to fill in the data into the format specified by Figure 21.

Evaluation of the hyper-plane location by orthogonal 2D Radon transforms

a) Principle

[0100] Maybe what scares the most in this approach of representing the Light-field is the necessity to work in a 4D space to locate the hyper-plane of ray-data. In the following analysis, we will show that we can evaluate the location and volume of the whole hyper-plane by just two orthogonal 2D DRTs.

[0101] Let us write down the equations of two orthogonal 2D sliced spaces from equation 2.

$$(z_2 - z_3)(x_1 + y_1) + (z_3 - z_1)(x_2 + y_2) = (z_2 - z_1)(x_3 + y_3)$$

[0102] And if we take a 2D slice for x_i coordinates, the equation of the beam of lines where ray data through an aperture of size A at (x_3,y_3,z_3) will map is:

$$x_2 = \frac{(z_3 - z_2)}{(z_3 - z_1)} x_1 + \frac{(z_2 - z_1)}{(z_3 - z_1)} (x_3 \pm A) = m x_1 + \left(d_{max_x} - d_{min_x}\right)$$

[0103] Similarly, if we take a 2D slice for y_i coordinates:

$$y_2 = \frac{(z_3 - z_2)}{(z_3 - z_1)} y_1 + \frac{(z_2 - z_1)}{(z_3 - z_1)} (y_3 \pm A) = m y_1 + \left(d_{max_y} - d_{min_y}\right)$$

[0104] According to the previous section, we can evaluate in the discrete domain the values of *m* and d_{max_x}, d_{min_x}, d_{max_y}, d_{min_y}.

[0105] To localize the characteristics of a light-field as defined by the format discussed previously, there is no need to perform a 4D discrete Radon transform. If we do two orthogonal 2D DRT, we can measure the slope m of the hyper-plane and the beam width of the digital hyper-planes where all data will concentrate in the 4D ray-diagram.

[0106] This simpler procedure of location assumes a circular entrance pupil A so that d_{max_x}, d_{min_x}, d_{max_y}, d_{min_y} will encompass all hyper-planes intercepts, some values written in the format will contain no values.

b) Example

[0107] Let us extend the example from section 3 to a 4D case. Let us assume that we have a camera located at (x_3,y_3,z_3) = (2,5,2). (i.e. the camera position is (x_3,y_3,z_3)) The 2D slice in the sampled ray diagram in x will be the same as the one represented on Figure 18, and the 2D-DRT is the one from Figure 23. The Figure 25 shows as well the orthogonal 2D-DRT in a y 2D-slice. More precisely, Figure 25 depicts two 2D ray diagrams: $\Pi(x_1,x_2)$, and $\Pi(y_1,y_2)$, as

well as the result of the Discrete Radon transform applied to these 2D ray diagrams.

[0108] From the 2D-xSlice DRT, we get an evaluation for the slope of the hyper-plane m = 0.5059 and from the 2D-ySlice DRT, another independent evaluation of m = 0,4980. From equations in section a), the true value is m = 0.5. The difference, which is small, comes from the fact that we transform a beam of digital lines, So that the maximum of the 2D Radon transform is broad and there is an uncertainty in the evaluation of the maximum. From the measurement of the waists, on both 2D-DRT, we find that d_{max_x} = 80, s = 208, d_{min_x} = 74, s = 201 for the beam envelope in x, and d_{max_y} = 87, s = 214, d_{min_y} = 80, s = 208 for the beam envelope in y. Those values are exactly what was expected.

[0109] It remains now to find a way to extend the format introduced in Figure 21 to the 4D context.

c) Case of multi-camera

[0110] If we have a light-field belonging to a multi camera system, we can proceed by taking advantage of the linearity of the DRT to estimate recursively the parametric volume that each camera occupies.

[0111] We can begin to do two orthogonal DRT and seek for the maximum and envelope of the digital hyper-planes belonging to that maximum. Once we have all the parameters, we can erase all the rays from the 4D ray-diagram or from the file where they are stored and begin again by a pair of orthogonal DRT to locate the second maximum, and so on until we have emptied the 4D ray-diagram.

[0112] We would be interested to obtain a format for the 4D case which is similar to what was proposed for the 2D case, i.e., table in Figure 21. To do so, we would be interested to associate the 2D lines found on the $\Pi(x_1,x_2)$ plane to the lines found on the $\Pi(y_1,y_2)$ plane, i.e., the lines that are the results of the intersection of the corresponding hyper plane with the two orthogonal slices of $\Pi(x_1,x_2)$ and $\Pi(y_1,y_2)$. From equations mentioned in section a), we know that the corresponding lines **<u>have the same slope</u>** m. This is the first parameter that associates each line on $\Pi(x_1,x_2)$ to a line in $\Pi(y_1,y_2)$ for a camera at a certain depth. Now, if we have multiple cameras at the same depth (i.e., the case of Figure 15), we have three lines in $\Pi(x_1,x_2)$ and three lines in $\Pi(y_1,y_2)$ with the same estimated slope ofm. We then need to determine the correspondences in the line offsets between the lines in these two planes. To do this, we exploit the formulation of the lines in Equations mentioned in section a). In particular, denoting $k = \frac{z_2 - z_1}{z_3 - z_1}$, we have the offsets as the following:

$$\begin{cases} kx_3 + kA = d_{max_x} \\ kx_3 - kA = d_{min_x} \end{cases}$$

and

$$\begin{cases} ky_3 + kA = d_{max_y} \\ ky_3 - kA = d_{min_y} \end{cases}$$

[0113] We can solve these sets of the equations for k, x_3 and y_3. Note that (x_3,y_3,z_3) are the coordinates of the camera, or in other words the center of projection, which is also the center of the exit pupil of radius A of the camera. We have supposed that the aperture on the plane positioned at z_3 is circular, so that $d_{max_x} - d_{min_x} = d_{max_y} - d_{min_y} = 2kA$, and by solving the previous sets of equations:

$$k = \frac{d_{max_x} - d_{min_x}}{2A}$$

$$x_3 = A\,\frac{d_{max_x} + d_{min_x}}{d_{max_x} - d_{min_x}}$$

$$y_3 = A\,\frac{d_{max_y} + d_{min_y}}{d_{max_y} - d_{min_y}}$$

$$z_3 = \frac{z_2 + (k-1)z_1}{k}$$

[0114] Now, we scan the digital lines the same as before on $\Pi(x_1,x_2)$ using the Bresenham digital lines. For each individual (x_1,x_2) value in $\Pi(x_1,x_2)$, we need to store the corresponding (y_1,y_2) value in $\Pi(y_1,y_2)$, captured in the light field. To find such values, we exploit equation 2. All the following are either known or estimated from previously mentioned equations.

[0115] Now moving on each line in $\Pi(x_1,x_2)$,, for each (x_1^q, x_2^q), we have the following relationship in (y_1,y_2):

$$y_2 = \frac{z_3 - z_2}{z_3 - z_1} y_1 + \frac{z_3 - z_2}{z_3 - z_1} x_1^q + \frac{z_2 - z_1}{z_3 - z_1}(x_3 + y_3{}^*) - x_2^q$$

[0116] Hence,

$$y_2 = m y_1 + m x_1^q + k(x_3 + y_3{}^*) - x_2^q = m y_1 + d_{off}(x_1^q, x_2^q, x_3, y_3{}^*)$$

[0117] It should be noted that in the case that we want to store lines parameter in $\Pi(y_1,y_2)$ in the format (instead of lines parameters of the lines identified in $\Pi(x_1,x_2)$), the same formula can be applied in order to recover from a point of a line in $\Pi(y_1,y_2)$ the corresponding value in $\Pi(x_1,x_2)$.

[0118] Where the aperture of the camera is a circle with the radius A. Therefore, variable y_3 *varies between $y_3 \pm A$. As a consequent, for each point in $\Pi(x_1,x_2)$, we need to save a collection of lines in $\Pi(y_1,y_2)$. **Therefore, $\boldsymbol{d_{off}(x_1^q, x_2^q, x_3, y_3{}^*)}$ corresponds to the offset of the lines that need to be scanned and saved for $\boldsymbol{(x_1^q, x_2^q)}$.**

To picture out how this works, we can refer to Figure 20. Each one of the colored squares is a (x_1^q, x_2^q), point, and for each one of these points, there is a set of Bresenham digital lines running out of the plane of the figure along a digital bundle defined by equation $y_2 = m y_1 + d_{off}(x_1^q, x_2^q, x_3, y_3{}^*)$ perpendicular to the depicted ones, but in a 4D space.

[0119] Now, the compact light field representation format is defined as follows (see Figure 22(a)): we first provide the general metadata of the 4D space: boundaries of the 4 axes, and their corresponding sampling. We then provide the number of cameras (bundles). For each camera, we create a block in the table, and we save the following parameters for camera *j* :

- cam_j corresponding to the size of aperture A_j;
- focusPoint =(x_3,y_3,z_3);
- lowest cam_j intercept in (x_1,x_2), noted d_j;
- m_j which correspond to cam_j steepness;
- l_j^x which corresponds to the number of digital lines in (x_1,x_2) for cam_j;
- l_j^y which corresponds to the number of digital lines in (y_1,y_2) for cam_j.

[0120] Then on this camera, for each (x_1^q, x_2^q), we start scanning (y_1,y_2). with respect to equation

$y_2 = my_1 + d_{off}(x_1^q, x_2^q, x_3, y_3{}^*)$ using the Bresenham digital lines, and we save the RGB values. In particular, $y_3{}^*$ is changed from $y_3{}^*$ - A to $y_3{}^*$ + A, and the corresponding d_{off} is calculated according to $mx_1^q + k(x_3 + y_3{}^*) - x_2^q$. The same calculations are performed in decoding sep using the stored metadata. In particular, k is found by determining $\frac{d_{max_x} - d_{min_x}}{2A}$. Hence, the format remains compact. We do not need to store four indexes for each ray in the system. Besides, note that we assume that the aperture size is known and the same for all the cameras. We however propose to save it for each camera to keep the format general.

[0121] It shall be noticed that the sampling of the hyper-plane above is the sampling of the 4D ray-space and we are sure not to miss a single x_1 y_1, x_2, y_2 location.

[0122] This is only an example of a systematic scanning of the 4D ray-space for saving all data into a very compact form. There could be other schemes of course. The parametric form seems to be adapted to explore the hyper-plane because it permits an inter-leaved space exploration.

Case of multi-cameras

[0123] To work on data that contain several bundles of hyper-planes (several maxima in the Radon transform due to multiple cameras), we can use a greedy algorithm. As a pre-processing step, the parameters (*m,k*) are found for all the peaks in the radon transform on $\Pi(x_1,x_2)$, and put in one set. The same is done for the peaks in (y_1, y_2) and the parameters are put in another set. Now in each iteration of the greedy algorithm, the maximum peak intensity is found in the 2D radon transform of (x_1,x_2), and the corresponding peak in (y_1,y_2), is found by matching the previously found parameters (*m, k*). After saving the data as mentioned in the last section, these peaks are cleaned from the radon transforms, and the next iteration is started, until nothing meaningful remains in the light field.

[0124] It should be noted that the context in Figure 22(a) is limited to one light slab. However, it can be generalized to the context when six light slabs are used. The Figures 22(c) and 22(d) depict an object (the penguin) that could be enclosed into a double box whose sides are used to build light slabs to parameterize the emitted light field capture by some randomly distributed cameras placed outside of the double box/cube. Hence, the file format of Figure 22(a) can be extended to hold tags for six light slabs. +z has already been assumed previously, but one can add -z, +x, -x, -y, +y to have a complete parametrization of the whole light field.

[0125] Each one of the six orientations gets then assigned a coordinate system for referencing the parametrization planes. Let us assume that we want to parametrize the direction $w \in$ {-z, +z, +x, -x, -y, +y}. This direction gets assigned the coordinate system ($\vec{e}_1$, $\vec{e}_2$, $\vec{e}_3$) with components (*u, v, w*). The planes are set at w_1 and w_2 and their dimensions are u_{min}, u_{max} and v_{min}, v_{max}. The 4D ray space will have components within u_1, u_2, v_1, v_2.

[0126] The Figure 22(b) presents the generalization of the Format in Figure 22(a). The Figure 22(b) presents only two light slab, but in general, depending of the application and on the scene also, there can be any number of section between one and six included. The value one would be the format for a light field propagation only in one direction, and the value of six would be for a light field propagation in all directions. It should be noted that the planes used to parametrize the light field are not necessary faces of the cube/box, and each one of the plane does not need to have a limitation of its size so as to build up a convex cube.

[0127] Figure 26(a) presents some steps of a method for providing a compact light field representation, according to one embodiment of the disclosure.

[0128] In one embodiment of the disclosure, a light field content is provided to an electronic device. Such light field content can be for example 4D light-field data as detailed previously or a two planes parametrization of light rays.

[0129] In a step referenced 2000, if the light field content is not represented by the parametrization of light rays via the use of two planes, a parametrization of light rays of the light field content is done via the use of two planes. The Figure 26(b) presents a synthetic scheme that depicts a conversion from either 4D light-field data or a conventional 2D image into a representation based on the parametrization of light rays via the use of two planes, corresponding to said step 2000..

[0130] Then, in a step referenced 2001, projections of light rays on one plane is done (either on the plane 405 or the plane 404, and generation of one 2D ray diagram is done (i.e. the 2D ray diagram $\Pi(x_1,x_2)$ in the case that projection on plane 404 is used, or the 2D ray diagram $\Pi(y_1,y_2)$ in the case that projection on plane 405 is used. Obviously, the generation of said one 2D ray diagram can be obtained directly from the sampling of the two parametrization planes.

[0131] In a step referenced 2002, the discrete Radon transform is applied on the 2D ray diagram outputted in step 2001 (i.e. the discrete Radon Transform is applied on either $\Pi(x_1,x_2)$ or on $\Pi(y_1,y_2)$ depending of the output of step

2001) in order to identify some lines of interest in the 2D ray diagram.

[0132] In a step referenced 2003, the Bresenham algorithm is used in order to encode the identified lines in the step 2002. It delivers for example for an identified line parameters d and s. In a variant, it only delivers parameters d, and the slope of the identified line. In the case that several identified lines belong to a beam (as explained previously), the slope and the thickness of the beam can be stored.

[0133] In a step referenced 2004, the light field content is stored in a compact way. Indeed, as it is possible to associate identified lines in step 2002 with lines in another 2Dray diagram, it is not necessary to store the coordinates in said another 2D ray diagram. Indeed, in the case that the outputted 2D ray diagram in step 2001 corresponds to the 2D ray diagram $\Pi(x_1,x_2)$, as we know that an identified line in $\Pi(x_1,x_2)$ must be associated with a line with the same slope in the 2D ray diagram $\Pi(y_1,y_2)$, and more precisely the following equation stands: $y_2 = my_1 + mx_1^q + k(x_3 + y_3{}^*) - x_2^q = my_1 + d_{off}(x_1^q, x_2^q, x_3, y_3{}^*)$, then, for each point on an identified line (i.e. the point with coordinates x_1^q, x_2^q, which does not have to be explicitly stored as it belongs to an identified line), a value (i.e. an RGB value) of the corresponding ray light passing through a point of the collection of points $(y_1, my_1 + d_{off}(x_1^q, x_2^q, x_3, y_3{}^*))$ is stored. However, in some case, the corresponding ray light does not exist, and a null value is stored. For example, taking the values in Figure 19, which presents an identified line, then when an electronic have to store the light field content, the value (i.e. RGB value) of the light ray passing though coordinates $x_1 = 0$, $x_2 = 5$, $y_1 = 0$, and $y_2 = m.0 + d_{off}(0,5,x_3,y_3{}^*)$ will be stored, as well as the value (i.e. RGB value) of the light ray passing though $x_1 = 0, x_2 = 5$, $y_1 = 1$, and $y_2 = m.1 + d_{off}(0,5,x_3,y_3{}^*)$, etc. until the value of y_2 is out of the boundaries of the planes (for example, the value (i.e. RGB value) of the light ray passing though $x_1 = 0, x_2 = 5$, $y_1 = 15$, and $y_2 = m.15 + d_{off}(0,5,x_3,y_3{}^*)$, is not stored as it is out of the boundaries of the plane. Then, the value (i.e. RGB value) of the light ray passing though coordinates $x_1 = 1$, $x_2 = 6$, $y_1 = 0$, and $y_2 = m.0 + d_{off}(1,6,x_3,y_3{}^*)$ will be stored, as well as the value (i.e. RGB value) of the light ray passing though $x_1 = 1$, $x_2 = 6$, $y_1 = 1$, and $y_2 = m.1 + d_{off}(1,6,x_3,y_3{}^*)$, etc. until the value of y_2 is out of the boundaries of the planes. The same process is repeated for each point on the identified line (i.e. until the processing of point $x_1 = 15$, $x_2 = 13$). The light field is encoded in this compact format when all the points belonging to lines identified in step 2002 have been handled.

[0134] The Figure 26(b) presents a synthetic scheme that depicts a conversion from either 4D light-field data or a conventional 2D image into a representation based on the parametrization of light rays via the use of two planes.

[0135] In one embodiment, it should be noted that it is possible to obtain from a conventional camera (i.e. from a conventional 2D image comprising a set of pixels), a set of light rays that correspond to a rough/average light field representation via a pixel back tracing operation.

[0136] In one embodiment of the disclosure, inputted 4D light-field data (obtained via a plenoptic camera or a camera array for example) comprise a set of pixels from which a collection of rays can be extracted. In order to extract such collection of rays, the method for converting comprises a pixel back tracing method/operation that can convert a pixel into a light ray. Indeed, as it is possible to convert one of the representation of 4D light field data into a set of sub-aperture images (i.e. a set of images from different point of views, taken at a same or close time. Each images can be virtually associated with a pixel sensor), therefore by applying a pixel back tracing on each images comprised in the set of sub-aperture images, a ray representation of the 4D light field data can be obtained.

[0137] Once a set of rays has been extracted from the pixels (or virtual pixels), a parametrization operation is executed by an electronic device. Such parametrization operation comprises the use of two planes (obviously, these planes are finite plane) in order to represent a ray as 4 coordinates (i.e. the coordinates of the intersection of a ray with the two planes).

[0138] In the following, details related to the pixel back tracing operation from a "conventional" 2D image are given (then this pixel back tracing operation is applied on several (or all) images comprised in the set of sub-aperture images, in order to obtain a ray representation for 4D light field data as explained previously). We will suppose for the sake of simplicity that the center of projection of the camera is at (0, 0, 0), the image plane centered at (0, 0, f). The camera is not rotated, its principal axis is pointing in the Z direction. We also suppose that the pixels are squares so that $m_u = m_v = m$ pixels.meter^{-1}. Finally, the image plane is translated by (t_u,t_v) in such a way that the bottom left pixel is at ($u = 0$, $v = 0$). The pixels are addressed by their column indexes pair (i,j). If I is the image position vector in pixels $(i,j,1)$, then the relation between the pixel and a point $P(X,Y,Z)$ that maps to that pixel through the central projection is:

$$\begin{pmatrix} i \\ j \\ 1 \end{pmatrix} = \begin{pmatrix} mf & 0 & mt_u \\ 0 & mf & mt_v \\ 0 & 0 & 1 \end{pmatrix} . \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \Leftrightarrow I = KP$$

[0139] And now, m, f, t_u, t_v being known from a calibration step, back tracing the pixel leads simply to:

$$K^{-1}I = P$$

[0140] P is a point in 3D space where a ray launched from pixel (i,j).would pass through. So we have one position, P, getting the direction is a matter of evidence, since that ray also passes through (0,0,0), the direction iis the vector $P(X,Y,Z)$.

[0141] This example illustrates how to back-trace a pixel, or in other words, how to go from pixels to a light-field.

[0142] In the case of processing of 4D light-field data, a multi-camera model can be used (i.e. to each of the sub-aperture images is associated a camera).

[0143] The parameterization is the process of identifying uniquely each ray with some tags. As a ray passes through a position and has a direction, the most straightforward parameterization would be a set of 6 real numbers (x,y,z,u,v,w) mixing spatial and directional coordinates. For the sake of simplicity we will adopt here a two plane parameterization already discussed previously. The question which is very often asked is where to put those planes with respect to the acquisition system. Often in the literature, in particular for plenoptic cameras, for some calculation convenience, one is placed at the sensor, another one on the lenslets array or main lens. Those positions should be avoided, because micro-lenses have a thickness, so where exactly shall we put one of the planes? Main lenses have a very complicated lens prescription which is unknown. So the best is to put the planes outside of the acquisition system.

[0144] Let us assume we have one plane at $z = z_1$ and another one at $z = z_2$ with $z_2 > z_1$. we can find the parameters for a ray which has been back-traced from pixel (i,j) as:

$$\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \begin{pmatrix} mf & 0 & mt_u \\ 0 & mf & mt_v \\ 0 & 0 & 1 \end{pmatrix}^{-1} . \begin{pmatrix} i \\ j \\ 1 \end{pmatrix}$$

[0145] Once the system of equations is written, the two first lines have to be divided by z_1 as it is the rule for homogenous coordinates.

[0146] The second set of parameters which come from the intersection with the second plane can also be written as:

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = \begin{pmatrix} mf & 0 & mt_u \\ 0 & mf & mt_v \\ 0 & 0 & 1 \end{pmatrix}^{-1} . \begin{pmatrix} i \\ j \\ 1 \end{pmatrix}$$

[0147] As the couple (z_1,z_2) is known, we can see from the previous derivations that the ray can be uniquely described by the 4-uplet $\mathbb{R}^4$ In addition, the ray is assigned with a RGB value. So to represent the ray in this parameterization, we need 4 floats and 3 bytes, a total of 19 bytes (but we have not sampled the ray space so far)! In the image based format, the elemental data is represented by 2 floats and 3 bytes only. So we can see that this parameterization adds a lot of data to the content and in general, the light-field being a four dimensional data set, it needs an amount of storage which is not manageable as it is.

[0148] The Figure 26 (c) and Figure 26(d) present how a ray is "extracted" from a pixel (or virtual pixel) sensor. Indeed, from a pixel (or virtual pixel) referenced 261 or 264, a chief ray (referenced 262 or 263) can be obtained (being the ray passing through the peak of a cone (the cone being referenced 265)at the pixel location, to the center of the basis of the cone (the base being defined as a function of the size of the main lens of the camera).

[0149] Here, we will define a way to help the user to navigate through light-fields ray format. This is very tightly linked to the format description defined in previous sections. Maybe there are some ways to navigate in a multi-view capture (which is not a light-field), but what we present here is a way to take advantage of our specific format.

[0150] Figure 22(a) describes a specific compact ray format. In this format, the different cameras are stored in an organized way. Their position is tagged by x_{3i},y_{3i},z_{3i} triplets and we can decide to order cam_i first by depth z_{3i} from nearest to farthest, and then by x_{3i} from bottom to top, and finally y_{3i} also from bottom to top, relative to their axis orientation.

[0151] When a light-field is transmitted over a network or aired to an antenna, it can be transmitted only in part to save the bandwidth, so that one point of view only is transmitted. With that point of view, we can also transmit all the cam_i foci.

[0152] The viewer can then display onto his display, from a menu button, a 3D screen overlay which visualizes other light-field view-points available in the original format. This can be in the form of a 3D axis system showing in 3D where the other viewpoints are located relative to the actual viewer position.

[0153] With a directional pad, the user navigates to the desired position, validates the position, and a feedback is sent over a network to the server. The server then positions the chosen data line which has to be streamed or broadcasted to the end user.

[0154] The screen overlay can also vary in aspect and the navigation through the viewpoints.

[0155] The file format can be modified to tag rectified camera pairs so that the screen overlay shows up an indication also for stereo content in the case the user wants to watch a stereo content, in which case two slots of digital line data have to be transmitted.

[0156] What makes such technique possible in relation to the representation format is the particular light-field analysis. Because we know that hyper-planes which are closer have a smaller m than those which a farther, we can begin to sort the different data line according to the distance. As m is calculated by the discrete Radon transforms as described previously. Also, as mentioned previously, it is possible to compute x_{3i}, y_{3i} and, z_{3i} and sort the different data lines in the format according to the sign and modulus of the focus components.

[0157] The fact that the data lines are sorted according to their 3D focus position in volume, enables a quick access to the one which is needed for broadcasting. What is broad-casted is a view point, or a pair of viewpoints in the case of stereo, and mathematically, it is one particular bundle of hyper-planes (or a pair thereof), which is a collection of rays stored according to 4D Bresenham scan lines or parametric data scanning.

[0158] Figure 27 presents a menu that comprises a 3D representation from which a user can select a position of a camera associated with 4D light-field data.

[0159] More precisely, a display device referenced 270 presents to a user the position in a 3D space of several camera (positions referenced 271 to 275 correspond to the positions of several camera).

[0160] Then user via a remote control unit referenced 276 can select one of these position. Once the selection has been done, 4D light field data associated with the selected camera are obtained by said display device. Then, the display device displays such 4D light field data.

[0161] In another embodiment of the disclosure, the user can select one of these position by touching the display device (i.e. it is a kind of touchscreen) instead of using the remote control unit 276.

[0162] Moreover, in one embodiment of the disclosure, the displayed 3D representation from which a user can select a position of a camera associated with 4D light-field data corresponds to a bounded spatial zone. However, in some embodiments of the disclosure, the spatial zone comprising positions of cameras can be greater than the one displayed. Hence, it may be possible to display additional visual indicators (such as arrows, etc.) for indicating a direction where positions of cameras are available. In another embodiment, axis lines x, y, z, as well as an origin are displayed. In one embodiment, the position of the origin is centered according to user's input.

[0163] In one embodiment of the disclosure, it is possible to perform rotation of the three axis lines. In one embodiment of the disclosure, the user can zoom in or out in a selected zone chosen by the user.

[0164] Figure 28 presents an example of an electronic device that can be used to perform one or several steps of methods disclosed in the present document.

[0165] Such electronic device referenced 280 comprises a computing unit (for example a CPU, for "*Central Processing Unit*"), referenced 281, and one or more memory units (for example a RAM (for "*Random Access Memory*") block in which intermediate results can be stored temporarily during the execution of instructions a computer program, or a ROM block in which, among other things, computer programs are stored, or an EEPROM ("*Electrically-Erasable Programmable Read-Only Memory*") block, or a flash block) referenced 282. Computer programs are made of instructions that can be executed by the computing unit. Such electronic device 280 can also comprise a dedicated unit, referenced 283, constituting an input-output interface to allow the device 280 to communicate with other electronic devices. In particular, this dedicated unit 283 can be connected with an antenna (in order to perform communication without contacts), or with serial ports (to carry communications "contact"). It should be noted that the arrows in Figure 28 signify that the linked unit can exchange data through buses for example together.

[0166] In an alternative embodiment, some or all of the steps of the method previously described, can be implemented in hardware in a programmable FPGA ("*Field Programmable Gate Array*") component or ASIC ("*Application-Specific Integrated Circuit*") component.

[0167] In an alternative embodiment, some or all of the steps of the method previously described, can be executed on an electronic device comprising memory units and processing units as the one disclosed in the Figure 28.

[0168] In one embodiment of the disclosure, the electronic device depicted in Figure 28 can be comprised in a camera device that is configure to capture images (either conventional 2D images or a sampling of a light field). These images are stored on one or more memory units. Hence, these images can be viewed as bit stream data (i.e. a sequence of bits). Obviously, a bit stream can also be converted on byte stream and vice versa.

Claims

1. Method for rendering a 4D light-field data, said method being executed by an electronic device, and being **characterized in that** it comprises :

   obtaining at least two Cartesian coordinates, each Cartesian coordinate being associated with a camera position in a three dimensional space;
   obtaining at least one 4D light-field data associated with a selected Cartesian coordinates from said at least two Cartesian coordinates; and
   rendering obtained 4D light-field data.

2. Method for rendering according to claim 1, wherein obtaining said at least one 4D light-field data comprises receiving a digital lines representation of said at least one 4D light-field data.

3. Method for rendering according to any claims 1 to 2, wherein said obtaining at least one 4D light-field data comprises selecting data, based on said selected Cartesian coordinates, from a table.

4. Method for rendering according to claims 3, wherein said table is stored on distant server.

5. A computer-readable and non-transient storage medium storing a computer program comprising a set of computer-executable instructions to implement a method for processing 4D light-field data when the instructions are executed by a computer, that can configure the computer to perform the method of claims 1 to 4.

6. An electronic device for rendering 4D light-field data, said electronic device being **characterized in that** it comprises :

   a first module configured to obtain at least two Cartesian coordinates, each Cartesian coordinate being associated with a camera position in a three dimensional space;
   a second module configured to obtain at least one 4D light-field data associated with a selected Cartesian coordinates from said at least two Cartesian coordinates; and
   a third module configured to render obtained 4D light-field data.

7. An electronic device for rendering according to claim 6, wherein said second module is further configured to receive a digital lines representation of said at least one 4D light-field data.

8. The electronic device for rendering according to any claims 6 to 7, said second module is further configured to select data, based on said selected Cartesian coordinates, from a table.

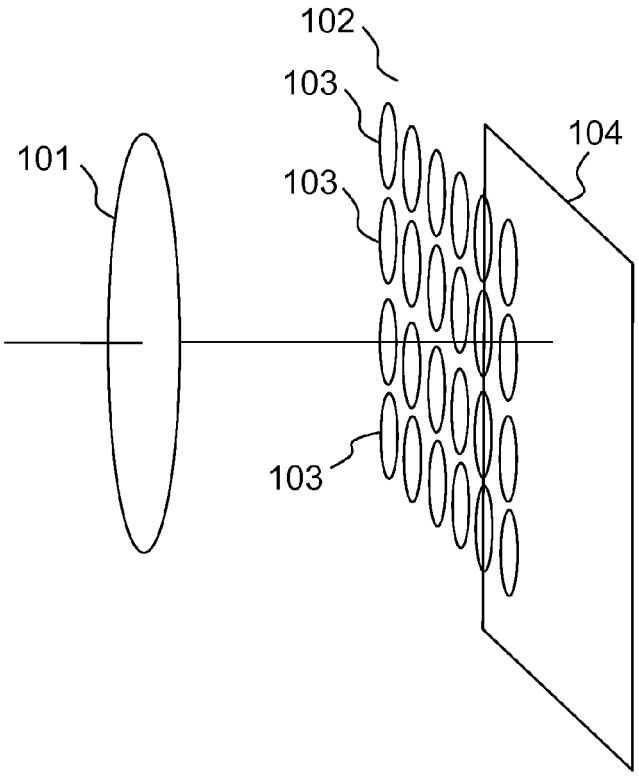

FIG.1

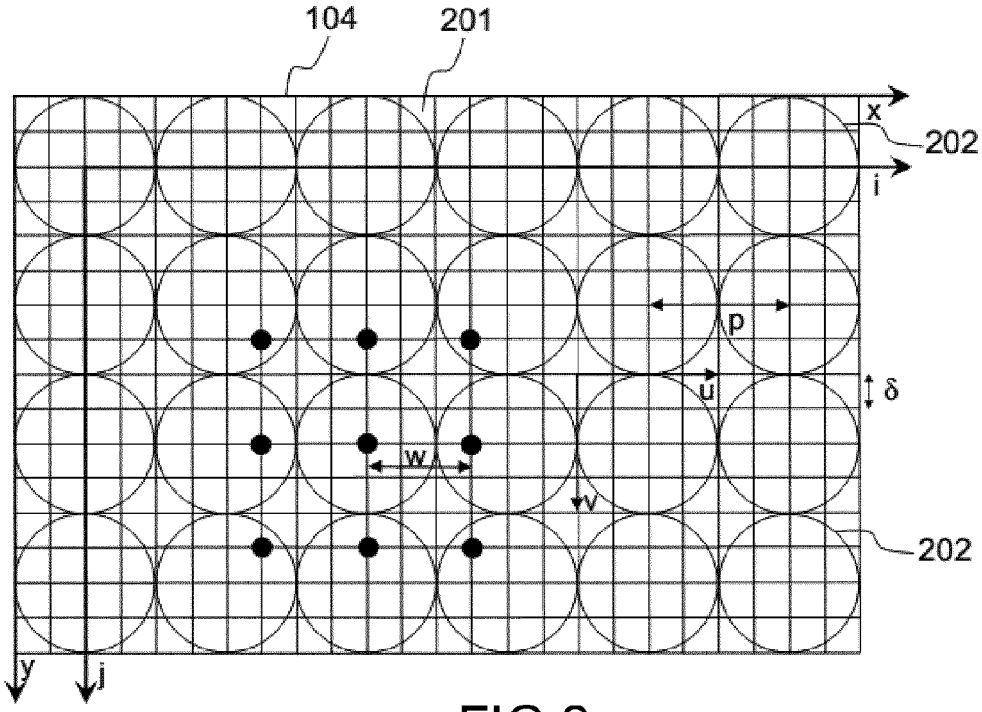

FIG.2

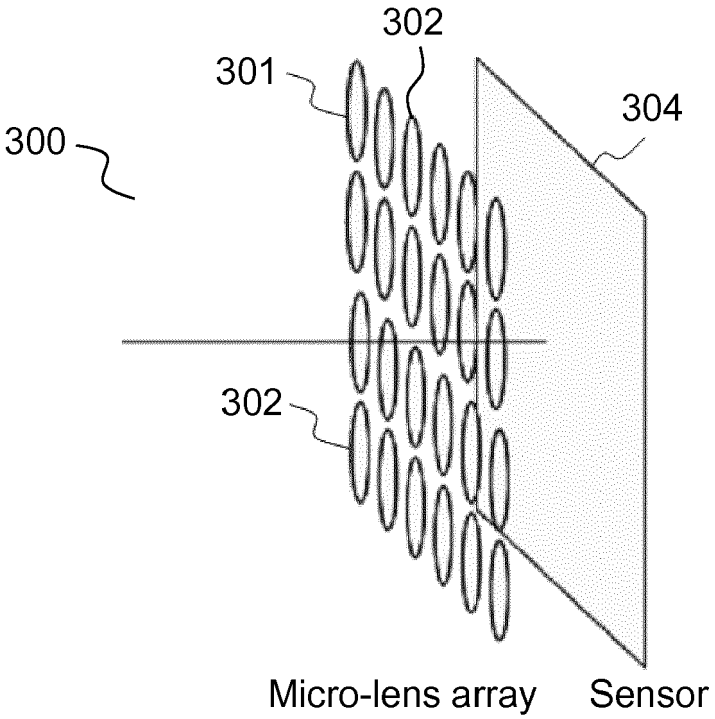

FIG.3

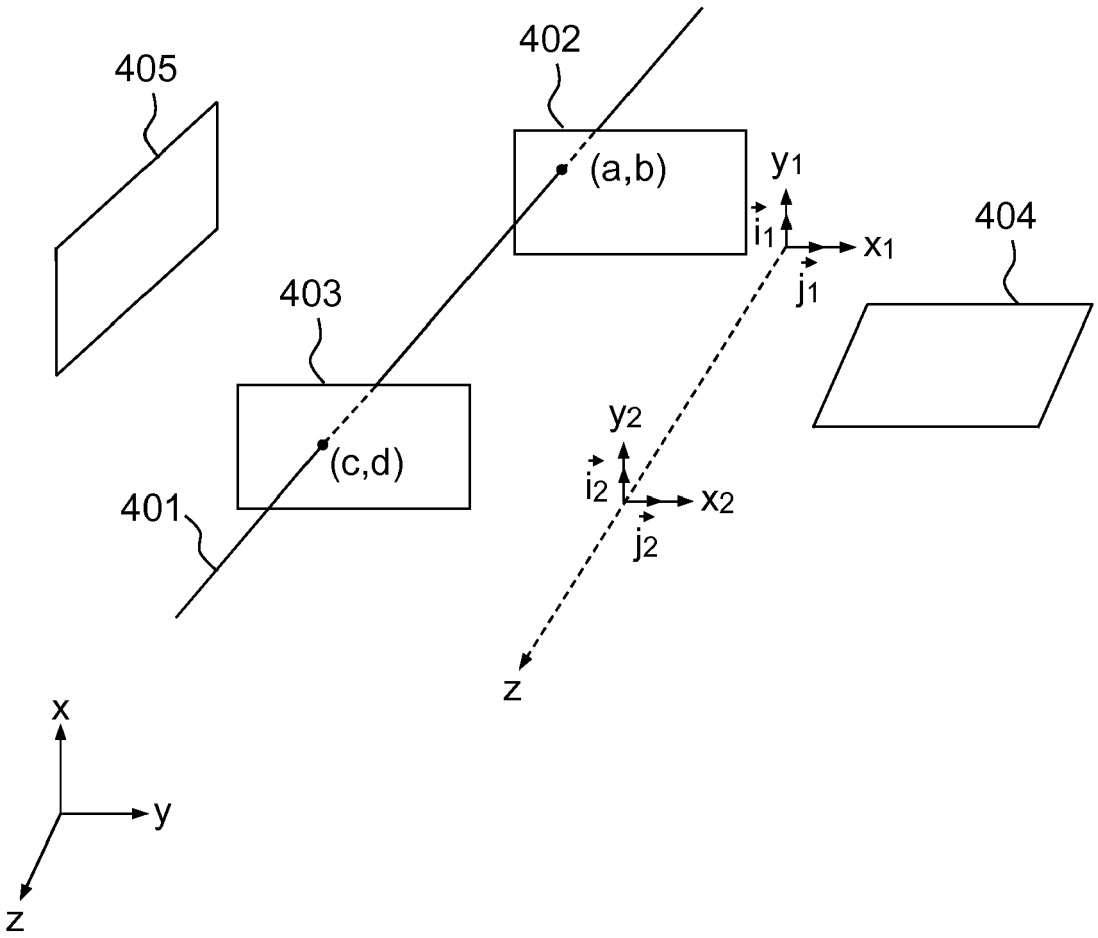

FIG.4

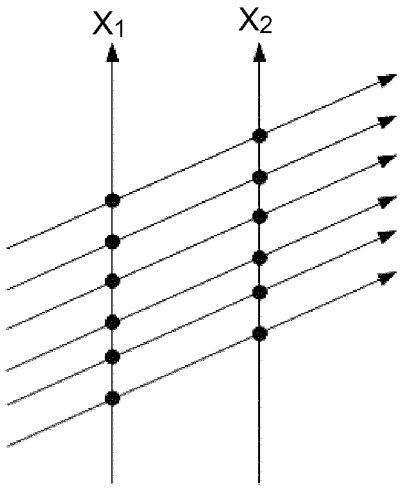

FIG.5(a)

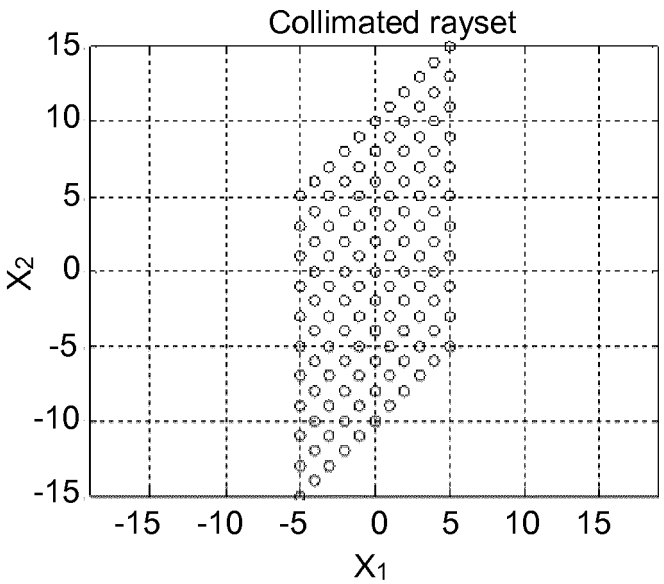

FIG.5(b)

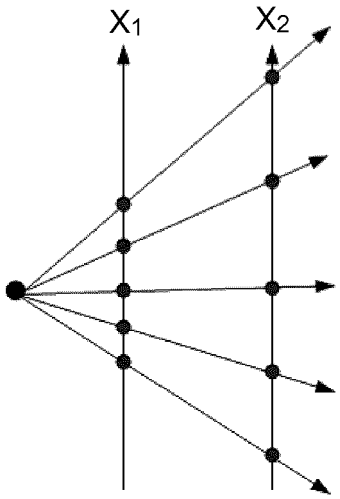

FIG.5(c)

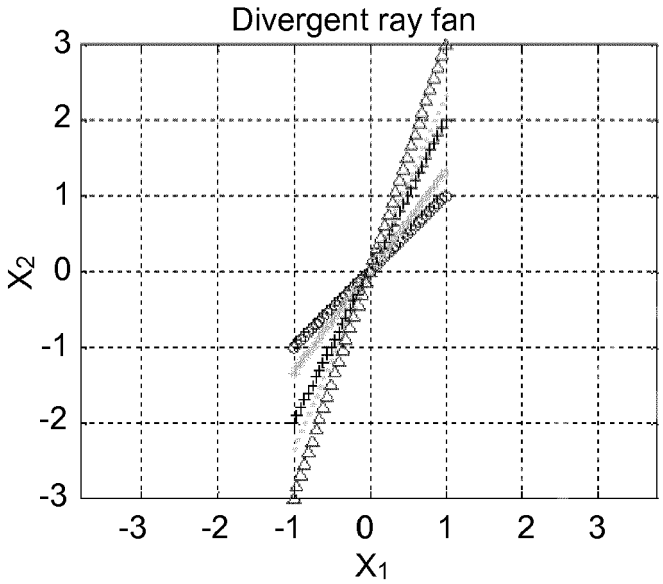

FIG.5(d)

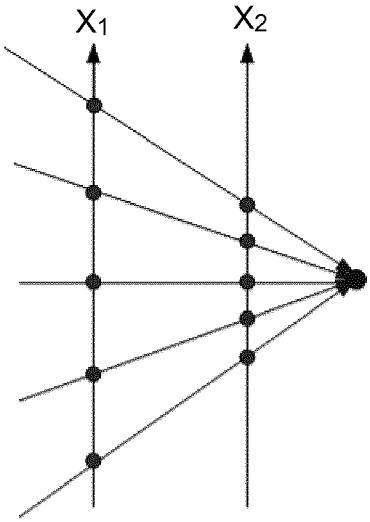

FIG.5(e)

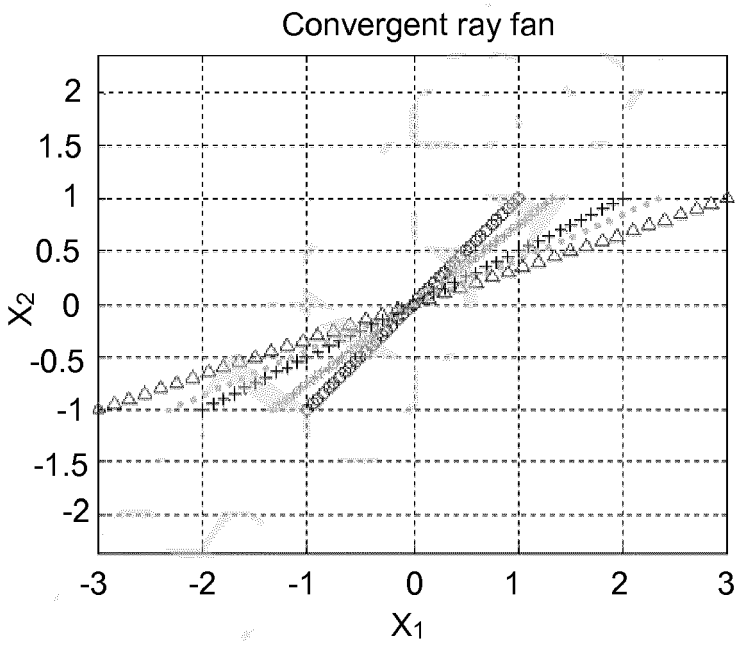

FIG.5(f)

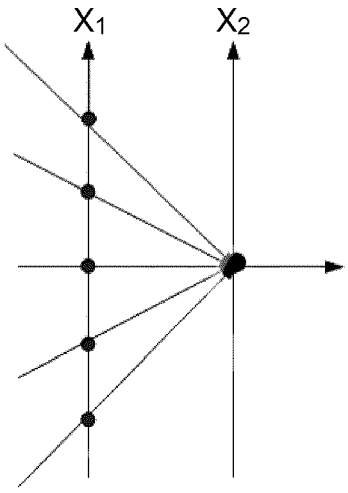

FIG.5(g)

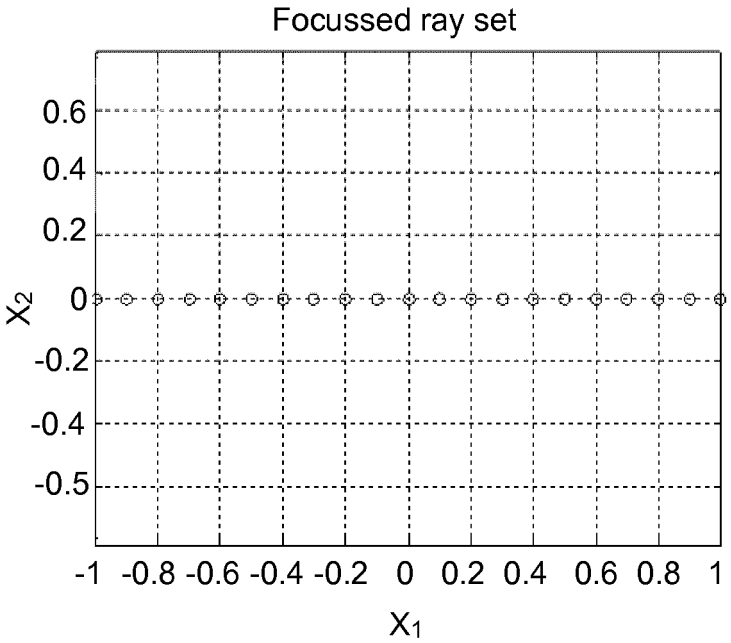

FIG.5(h)

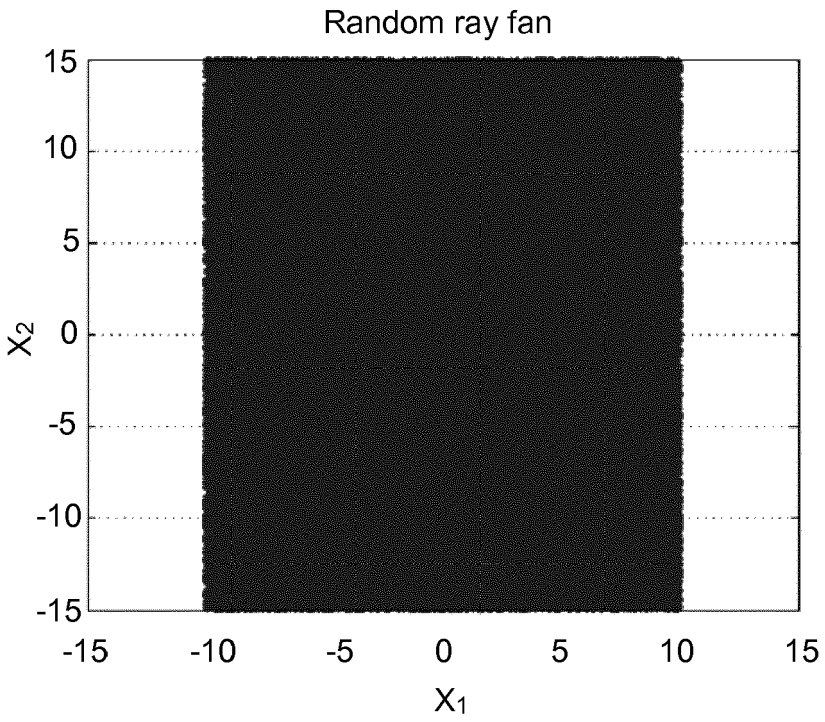
Random ray fan
15
10
5
0
-5
-10
-15
X2
-15
-10
-5
0
5
10
15
X1

FIG.6

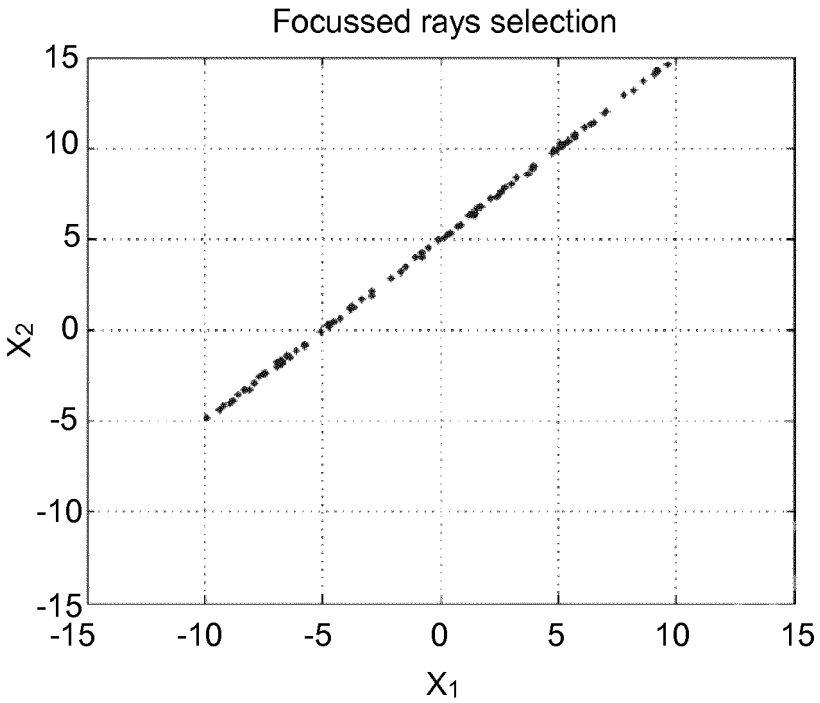
Focussed rays selection
15
10
5
0
-5
-10
-15
X2
-15
-10
-5
0
5
10
15
X1

FIG.7(a)

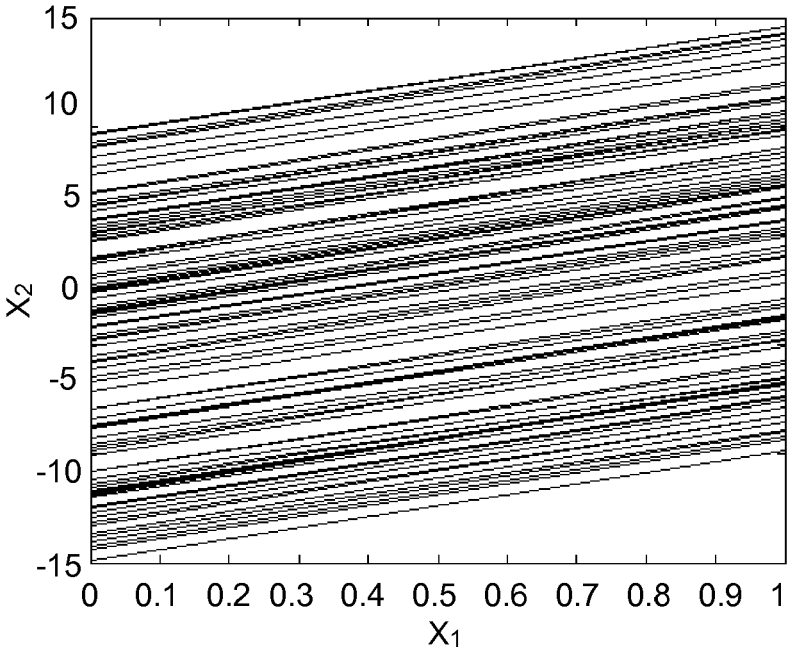
15
10
5
0
-5
-10
-15
X2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
X1

FIG.7(b)

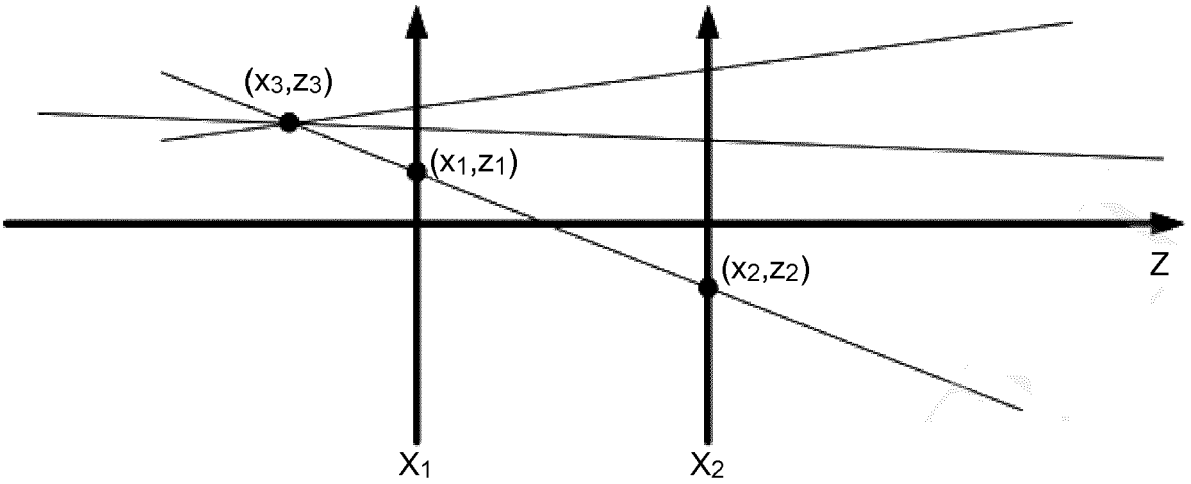
(x3,z3)
(x1,z1)
(x2,z2)
Z
X1
X2

FIG.8

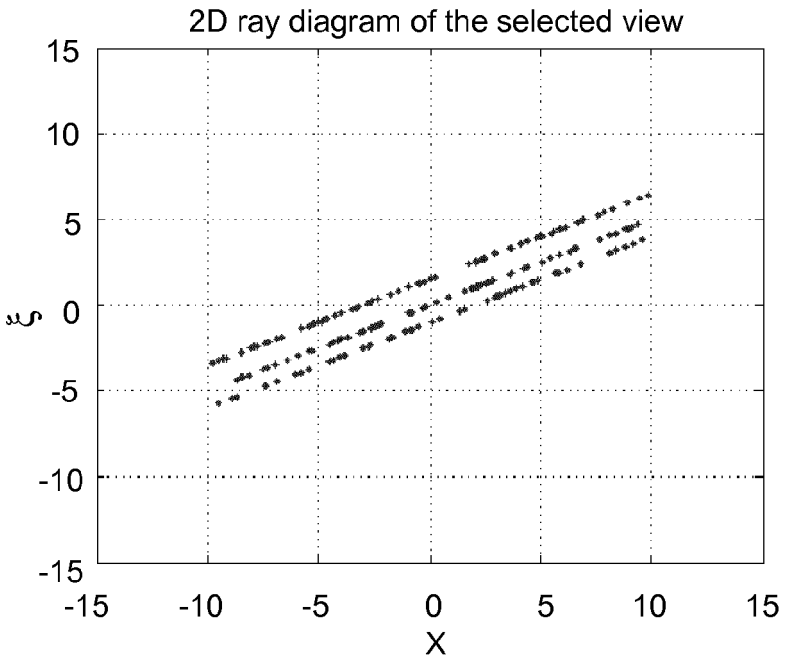
2D ray diagram of the selected view
15
10
5
0
-5
-10
-15
ξ
-15
-10
-5
0
5
10
15
X

FIG.9

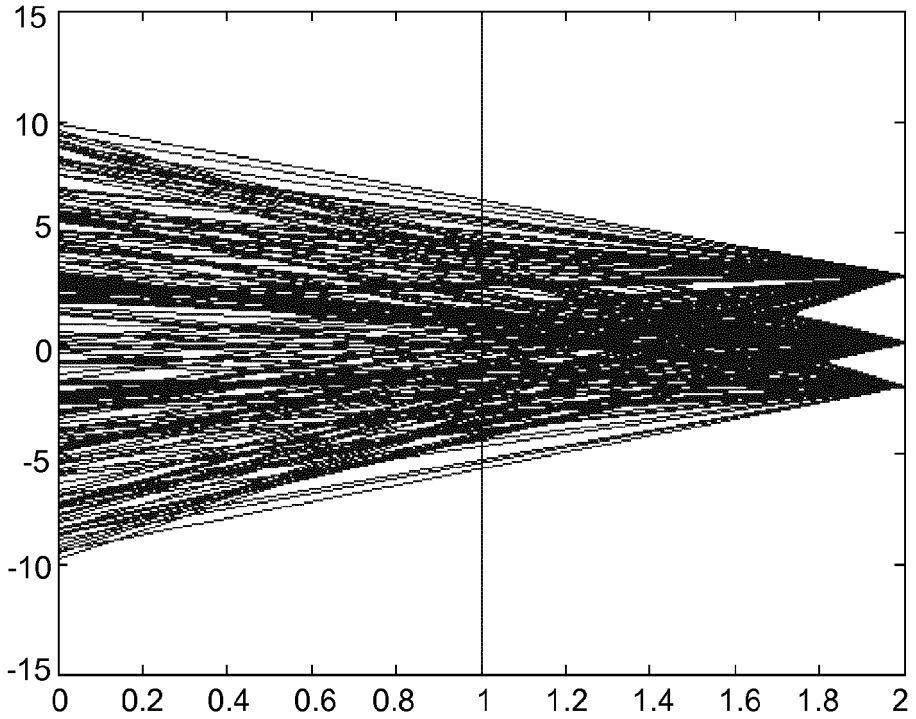
15
10
5
0
-5
-10
-15
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

FIG.10

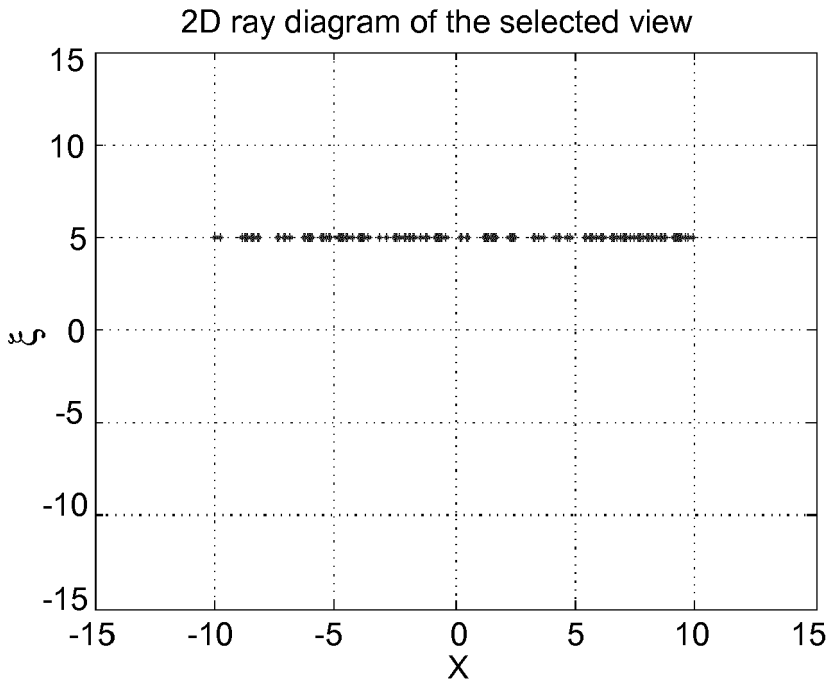
2D ray diagram of the selected view
15
10
5
0
-5
-10
-15
ξ
-15
-10
-5
0
5
10
15
X

FIG.11

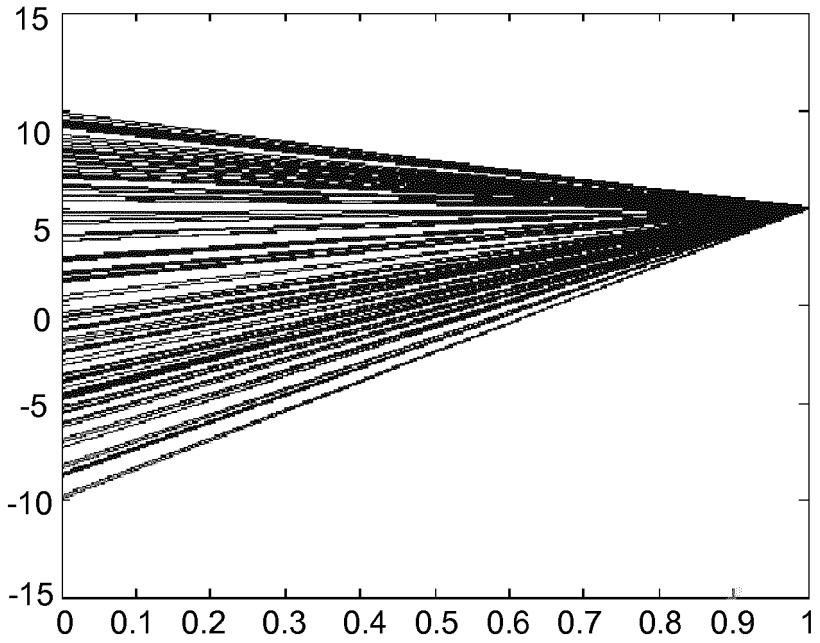
15
10
5
0
-5
-10
-15
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

FIG.12

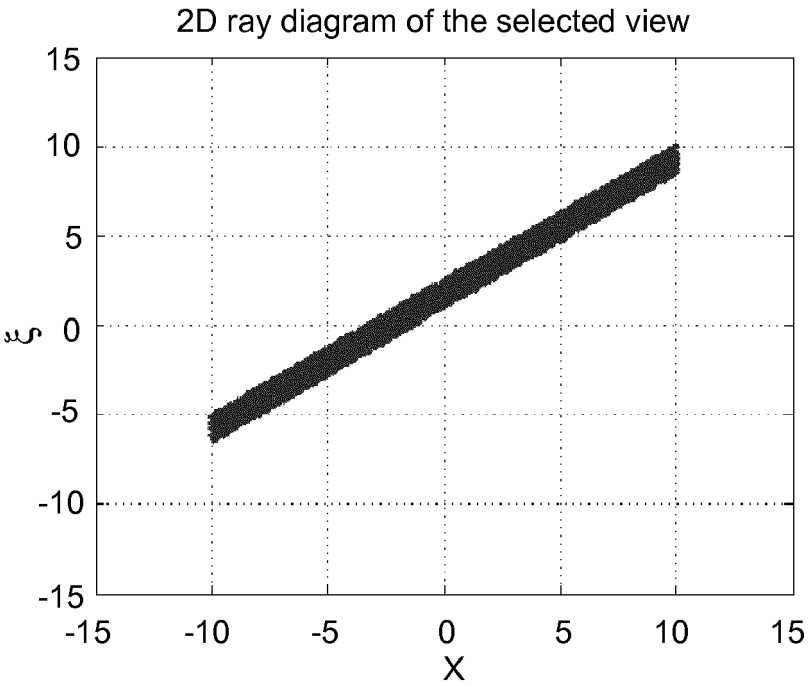
2D ray diagram of the selected view
15
10
5
0
-5
-10
-15
ξ
-15
-10
-5
0
5
10
15
X

FIG.13

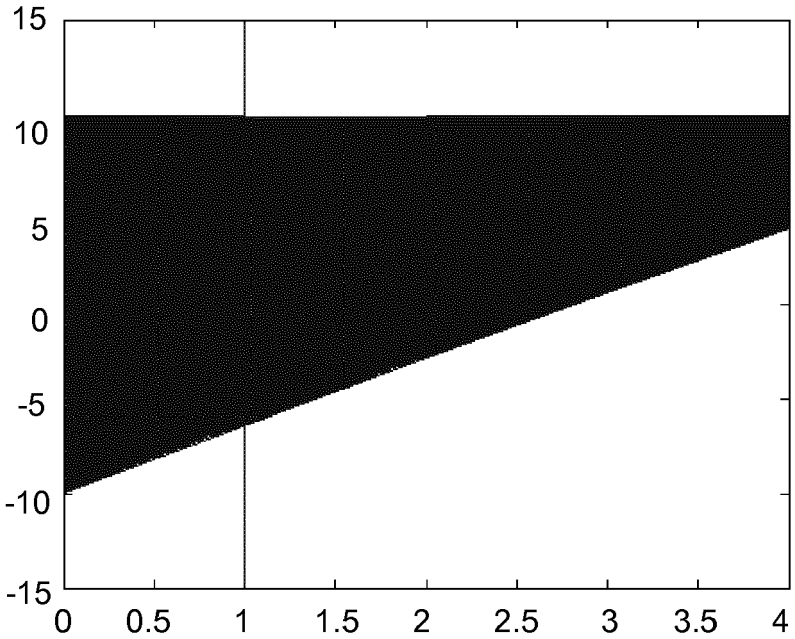
15
10
5
0
-5
-10
-15
0
0.5
1
1.5
2
2.5
3
3.5
4

FIG.14

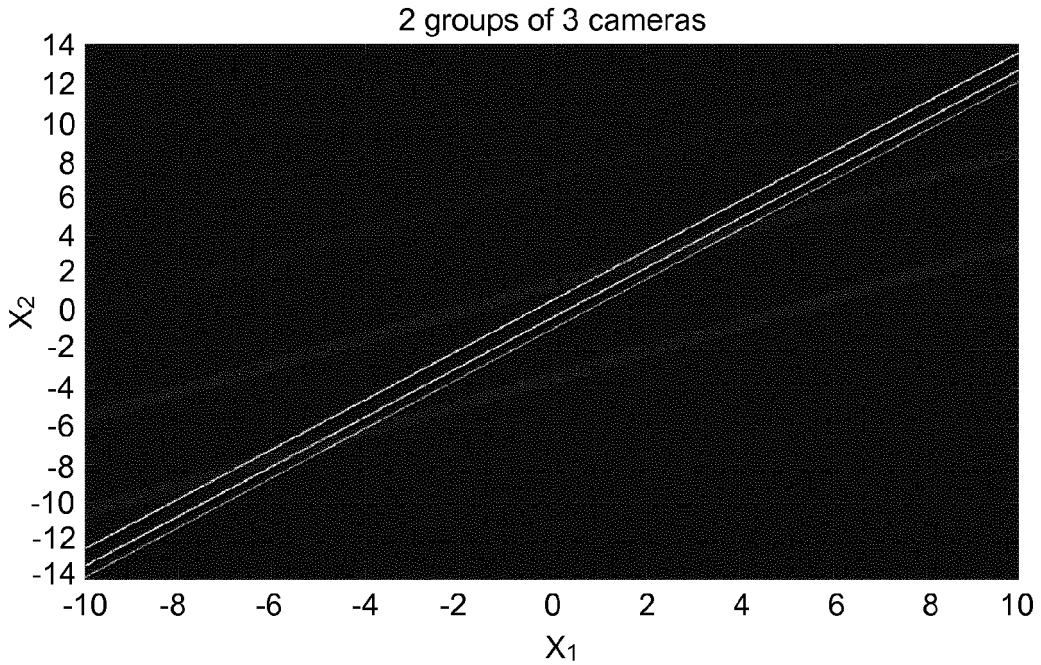
2 groups of 3 cameras
14
12
10
8
6
4
2
0
-2
-4
-6
-8
-10
-12
-14
X2
-10
-8
-6
-4
-2
0
2
4
6
8
10
X1

FIG.15

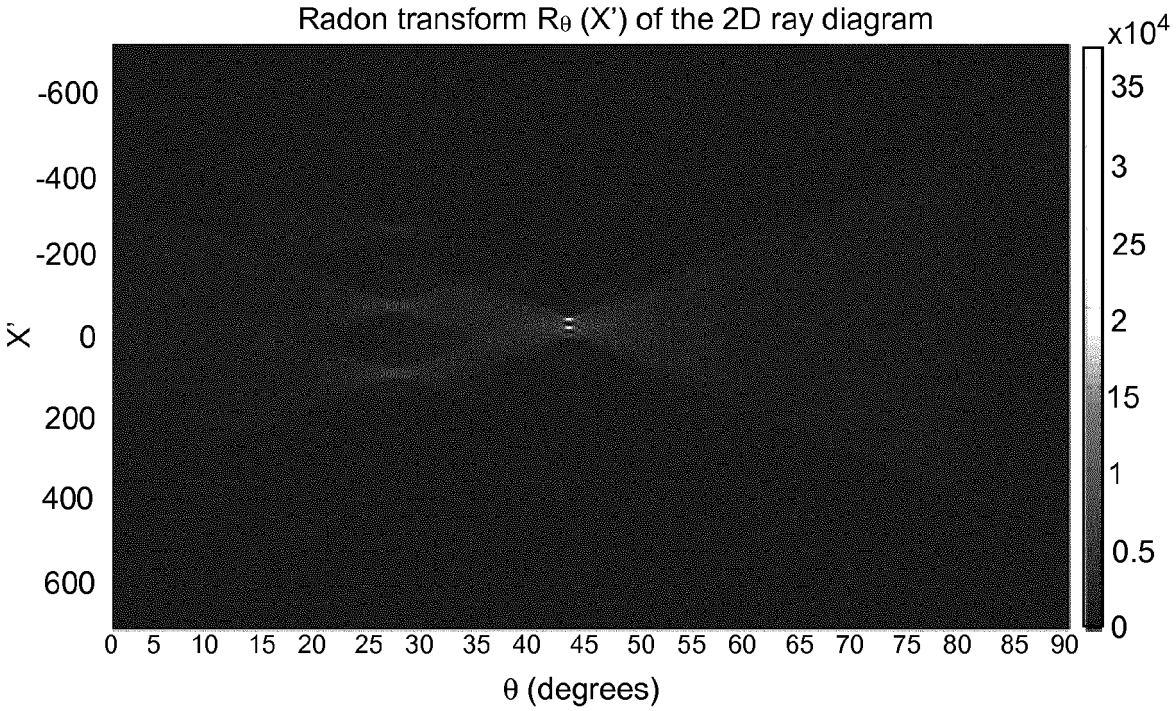
Radon transform Rθ (X') of the 2D ray diagram
-600
-400
-200
0
200
400
600
X'
0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90
θ (degrees)
x10^4
35
3
25
2
15
1
0.5
0

FIG.16

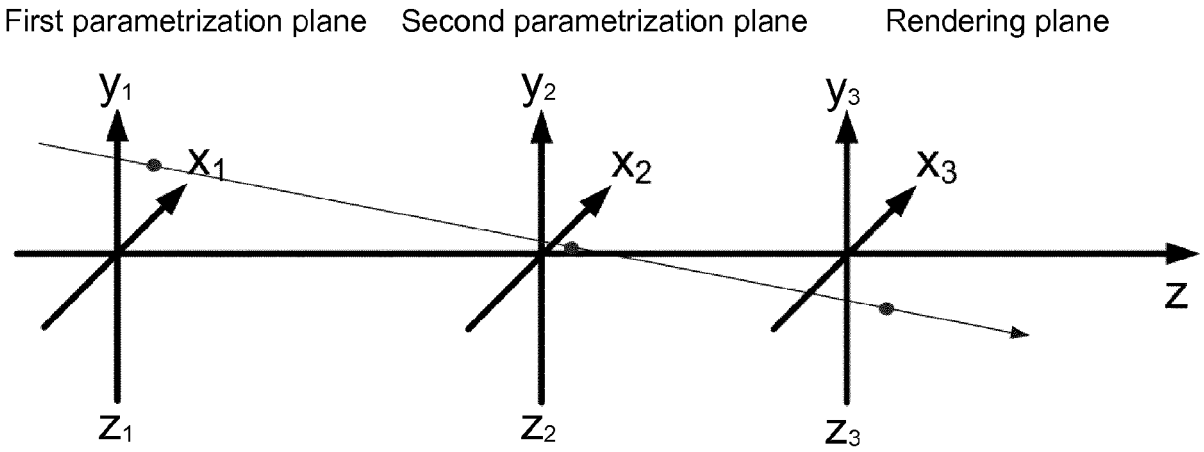
First parametrization plane
Second parametrization plane
Rendering plane
y1
y2
y3
x1
x2
x3
z
z1
z2
z3

FIG.17

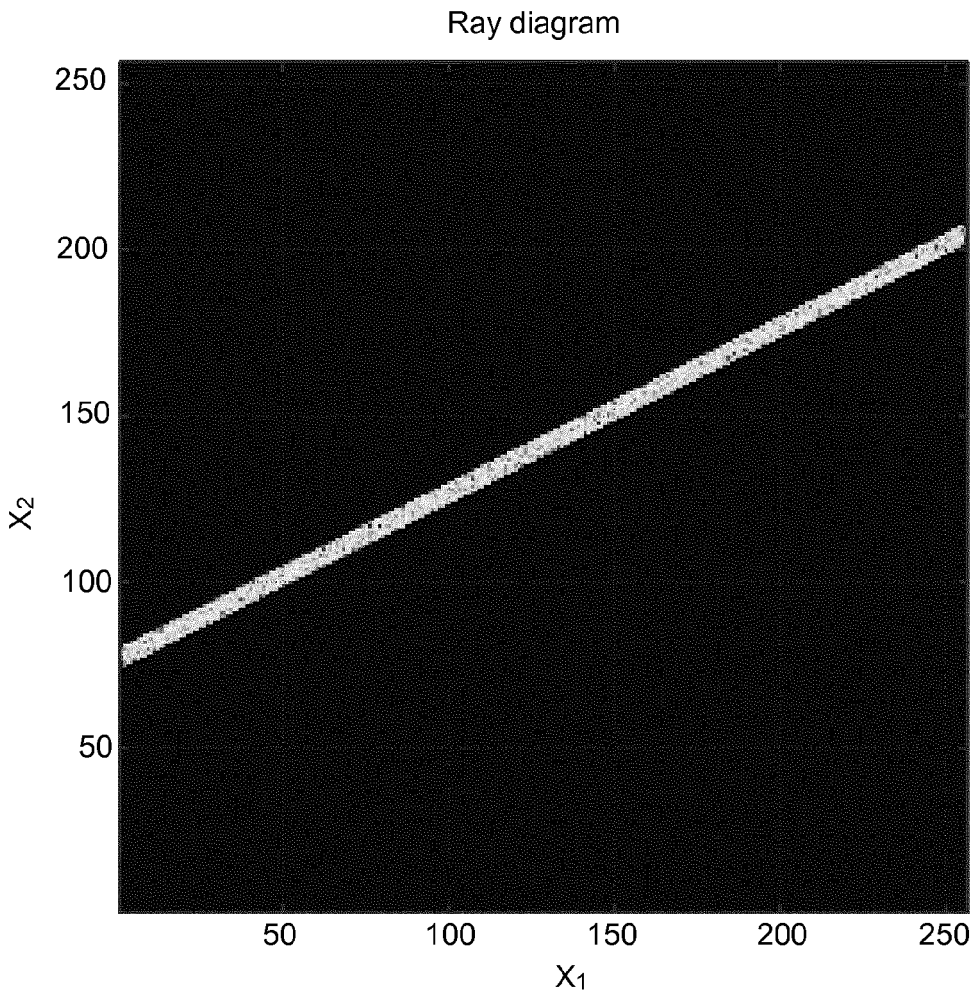
Ray diagram
250
200
150
100
50
x2
50
100
150
200
250
x1

FIG.18

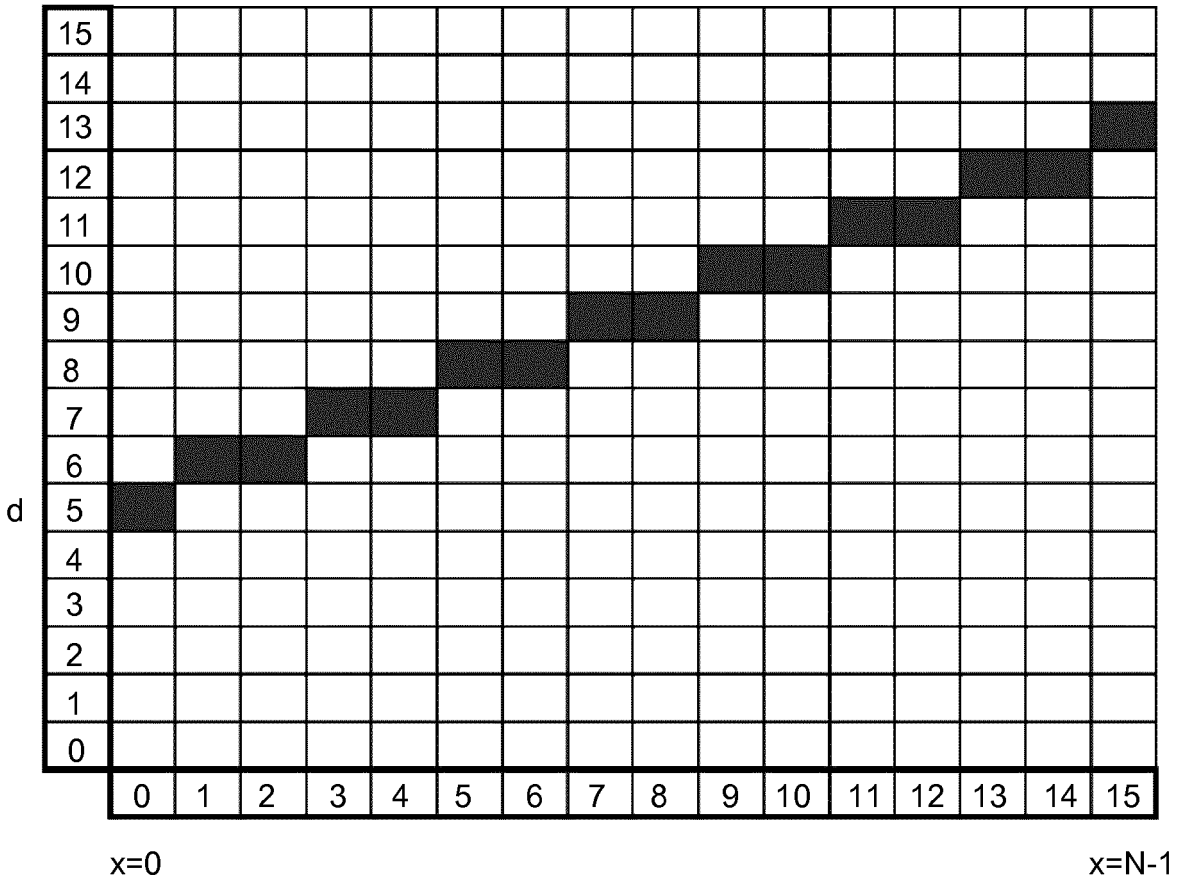
d
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
x=0
x=N-1

FIG.19

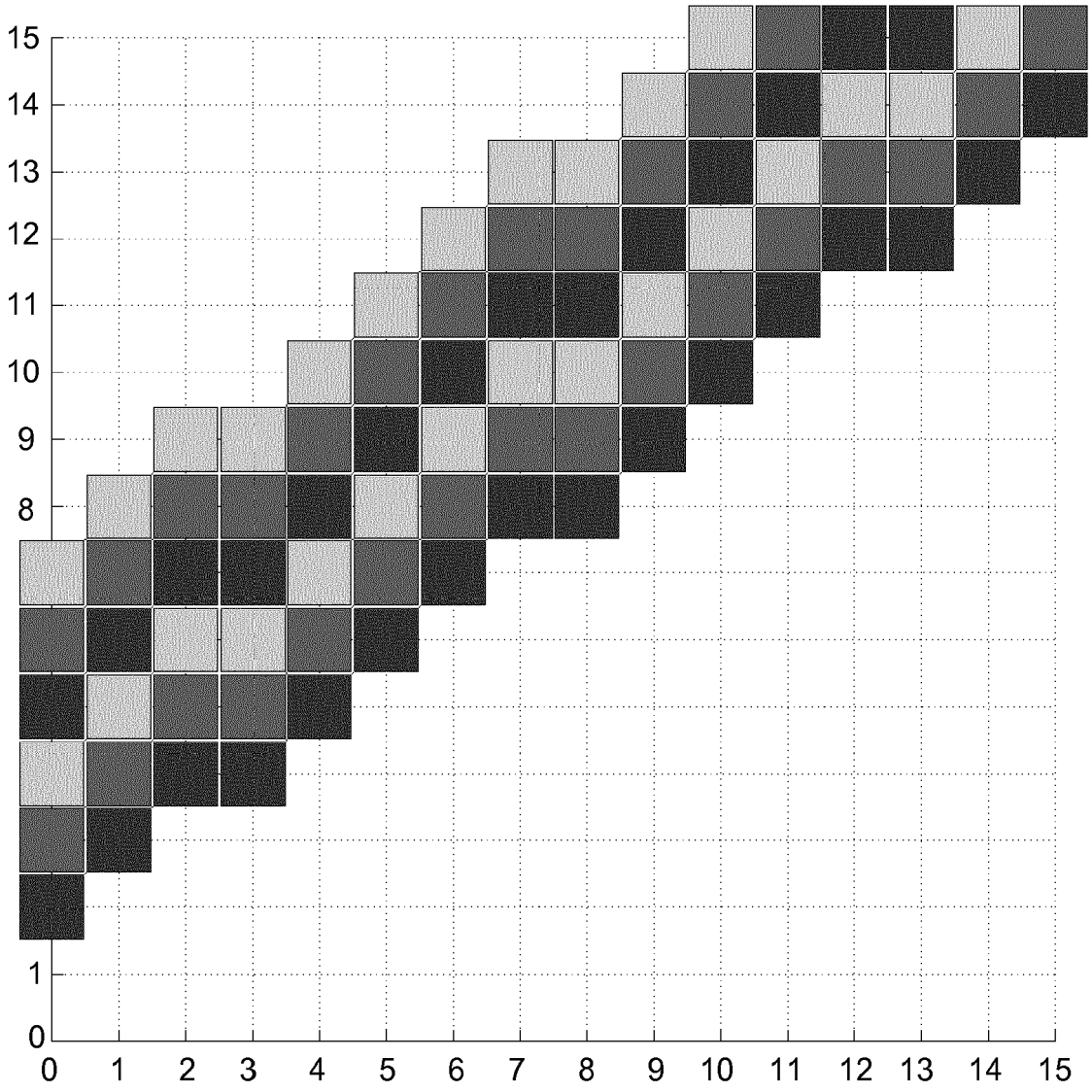

FIG.20

Tags			Data			
Parametrization plane offset z_2-z_1 (z_3 is relative to z_1) (float)			**Δz**			
Width in number of x_1 cells (integer)	min x_1 value (float)	max x_1 value (float)	**N**	$\mathbf{x_{1min}}$	$\mathbf{x_{1Max}}$	
Width in number of x_2 cells (integer)	min x_2 value (float)	max x_2 value (float)	**M**	$\mathbf{x_{2min}}$	$\mathbf{x_{2Max}}$	
Number of cameras (integer)			**n**			
Cam_1 focus (2 floats)			**x31**	**z31**		
lowest cam_1 intercept in (x_1,x_2)	cam_1 steepness	cam_1 number of digital lines in x_1,x_2	d_1	s_1	m(1)	
data line (m(1)*N * 3 bytes)			RGB_0	RGB_1	...	$RGB_{m(1)*N}$
...			...	...		
...			...	...	...	...
Cam_n focus (2 floats)			**x31**	**z31**		
lowest cam_n intercept in (x_1,x_2)	cam_n steepness	cam_n number of digital lines in x_1,x_2	d_n	s_n	m(n)	
data line (m(n)*N * 3 bytes)			RGB_0	RGB_1	...	$RGB_{m(n)*N}$

FIG.21

Tags			Data			
Parametrization plane offset z_2-z_1 (z_3 is relative to z_1) (float)			Δz			
Width in number of x_1 cells (integer)	min x_1 value (float)	max x_1 value (float)	**N**	x_{1min}	x_{1Max}	
Width in number of x_2 cells (integer)	min x_2 value (float)	max x_2 value (float)	**M**	x_{2min}	x_{2Max}	
Width in number of y_1 cells (integer)	min y_1 value (float)	max y_1 value (float)	**P**	y_{1min}	y_{1Max}	
Width in number of y_2 cells (integer)	min y_2 value (float)	max y_2 value (float)	**Q**	y_{2min}	y_{2Max}	
Number of cameras (integer)			**n**			
Aperture size for cam_1			A_1			
Cam_1 focus (3 floats)			$x3_1$	$y3_1$	$z3_1$	
lowest cam_1 intercept in (x_1,x_2)	cam_1 steepness	cam_1 number of digital lines in x_1,x_2	d_1	m_1	l^x_1	
		cam_1 number of digital lines in y_1,y_2			l^y_1	
data line (end_1=l^x_1*N*l^y_1*P) * 3 bytes			RGB_0	RGB_1	...	RGB_{end1}
...			...	...		
...			...	...	...	...
Aperture size for cam_n			A_n			
Cam_n focus (3 floats)			$x3_n$	$y3_n$	$z3_n$	
lowest cam_n intercept in (x_1,x_2)	cam_n steepness	cam_n number of digital lines in x_1,x_2	d_n	m_n	l^x_n	
		cam_n number of digital lines in y_1,y_2			l^y_n	
data line (end_n=l^x_n*N*l^y_n*P) * 3 bytes			RGB_0	RGB_1	...	RGB_{endn}

FIG.22(a)

Tags			Data			
Light slab orientation 1			one tag w from (-x,+x,-y,+y,-z,+z)			
Parametrization plane offset w_2-w_1 (w_3 is relative to w_1) (float)			Δe			
Width in number of u_1 cells (integer)	min u_1 value (float)	max u_1 value (float)	N	u_{1min}	u_{1Max}	
Width in number of u_2 cells (integer)	min u_2 value (float)	max u_2 value (float)	M	u_{2min}	u_{2Max}	
Width in number of v_1 cells (integer)	min v_1 value (float)	max v_1 value (float)	P	v_{1min}	v_{1Max}	
Width in number of v_2 cells (integer)	min v_2 value (float)	max v_2 value (float)	Q	v_{2min}	v_{2Max}	
Number of cameras (integer)			n			
Aperture size for cam_1			A_1			
Cam_1 focus (3 floats)			$u3_1$	$v3_1$	$w3_1$	
lowest cam_1 intercept in (u_1,u_2)	cam_1 steepness	cam_1 number of digital lines in u_1,u_2	d_1	m_1	l^x_1	
		cam_1 number of digital lines in v_1,v_2			l^y_1	
data line (end_1=l^x_1*N*l^y_1*P) * 3 bytes			RGB_0	RGB_1	...	RGB_{end1}
...			...	...		
...			...	...	...	...
Aperture size for cam_n			A_n			
Cam_n focus (3 floats)			$x3_n$	$y3_n$	$z3_n$	
lowest cam_n intercept in (u_1,u_2)	cam_n steepness	cam_n number of digital lines in u_1,u_2	d_n	m_n	l^x_n	
		cam_n number of digital lines in v_1,v_2			l^y_n	
data line (end_n=l^x_n*N*l^y_n*P) * 3 bytes			RGB_0	RGB_1	...	RGB_{endn}

FIG.22(b)

Light slab orientation 2			one tag w from (-x,+x,-y,+y,-z,+z)			
Parametrization plane offset w_2-w_1 (w_3 is relative to w_1) (float)			Δe			
Width in number of u_1 cells (integer)	min u_1 value (float)	max u_1 value (float)	**N**	u_{1min}	u_{1Max}	
Width in number of u_2 cells (integer)	min u_2 value (float)	max u_2 value (float)	**M**	u_{2min}	u_{2Max}	
Width in number of v_1 cells (integer)	min v_1 value (float)	max v_1 value (float)	**P**	v_{1min}	v_{1Max}	
Width in number of v_2 cells (integer)	min v_2 value (float)	max v_2 value (float)	**Q**	v_{2min}	v_{2Max}	
Number of cameras (integer)			**n**			
Aperture size for cam_1			A_1			
Cam_1 focus (3 floats)			$u3_1$	$v3_1$	$w3_1$	
lowest cam_1 intercept in (u_1,u_2)	cam_1 steepness	cam_1 number of digital lines in u_1,u_2	d_1	m_1	I^x_1	
		cam_1 number of digital lines in v_1,v_2			I^y_1	
data line (end_1=I^x_1*N*I^y_1*P) * 3 bytes			RGB_0	RGB_1	...	RGB_{end1}
...			...	...		
...			...	...	...	...
Aperture size for cam_n			A_n			
Cam_n focus (3 floats)			$x3_n$	$y3_n$	$z3_n$	
lowest cam_n intercept in (u_1,u_2)	cam_n steepness	cam_n number of digital lines in u_1,u_2	d_n	m_n	I^x_n	
		cam_n number of digital lines in v_1,v_2			I^y_n	
data line (end_n=I^x_n*N*I^y_n*P) * 3 bytes			RGB_0	RGB_1	...	RGB_{endn}

FIG.22(b)

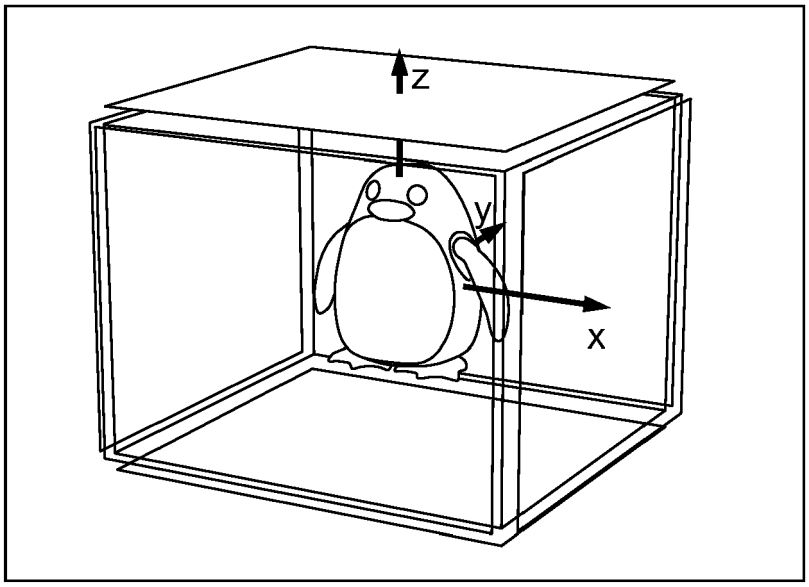

FIG.22(c)

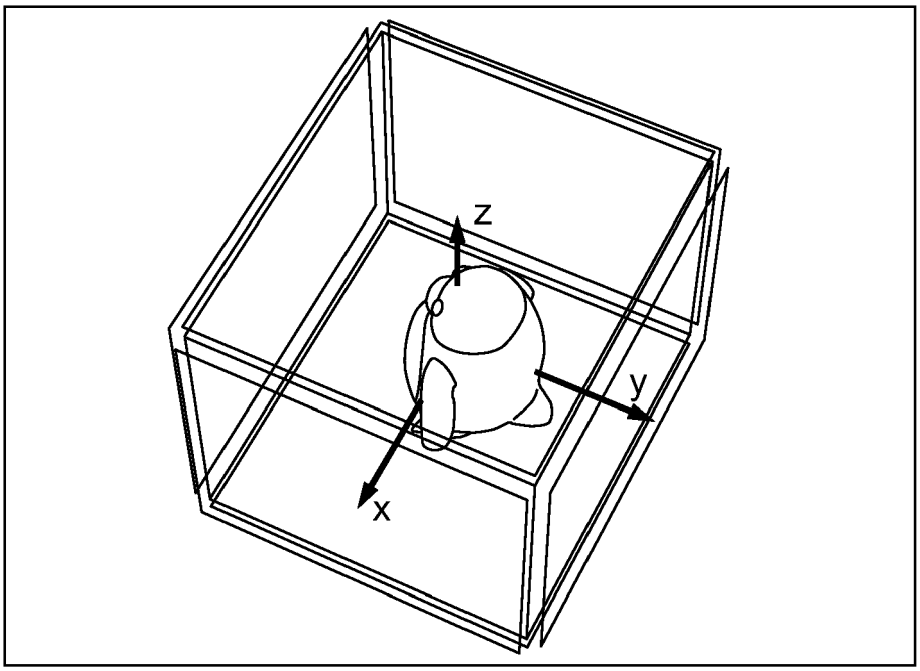

FIG.22(d)

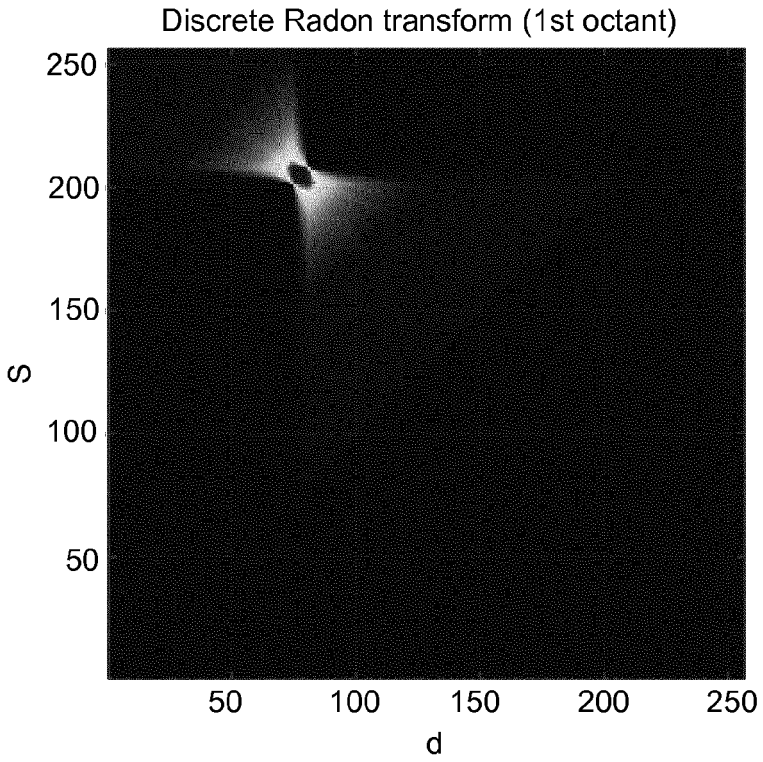
Discrete Radon transform (1st octant)
250
200
150
100
50
s
50
100
150
200
250
d

FIG.23

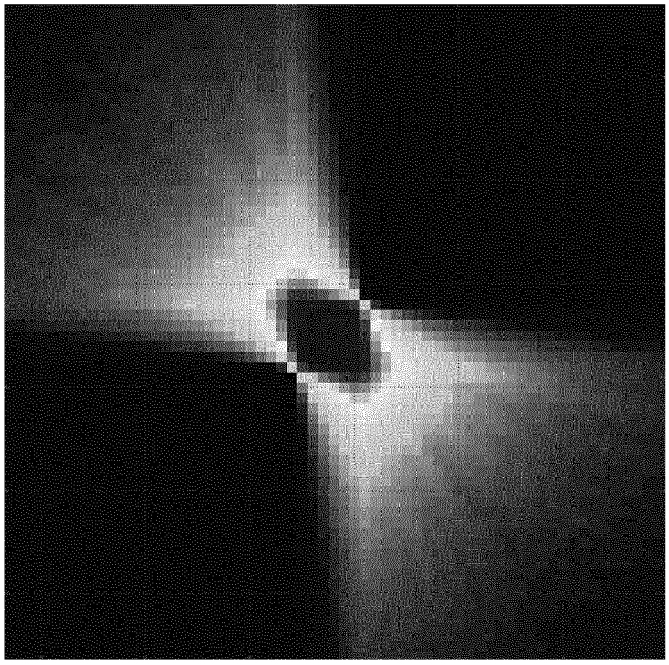

FIG.24

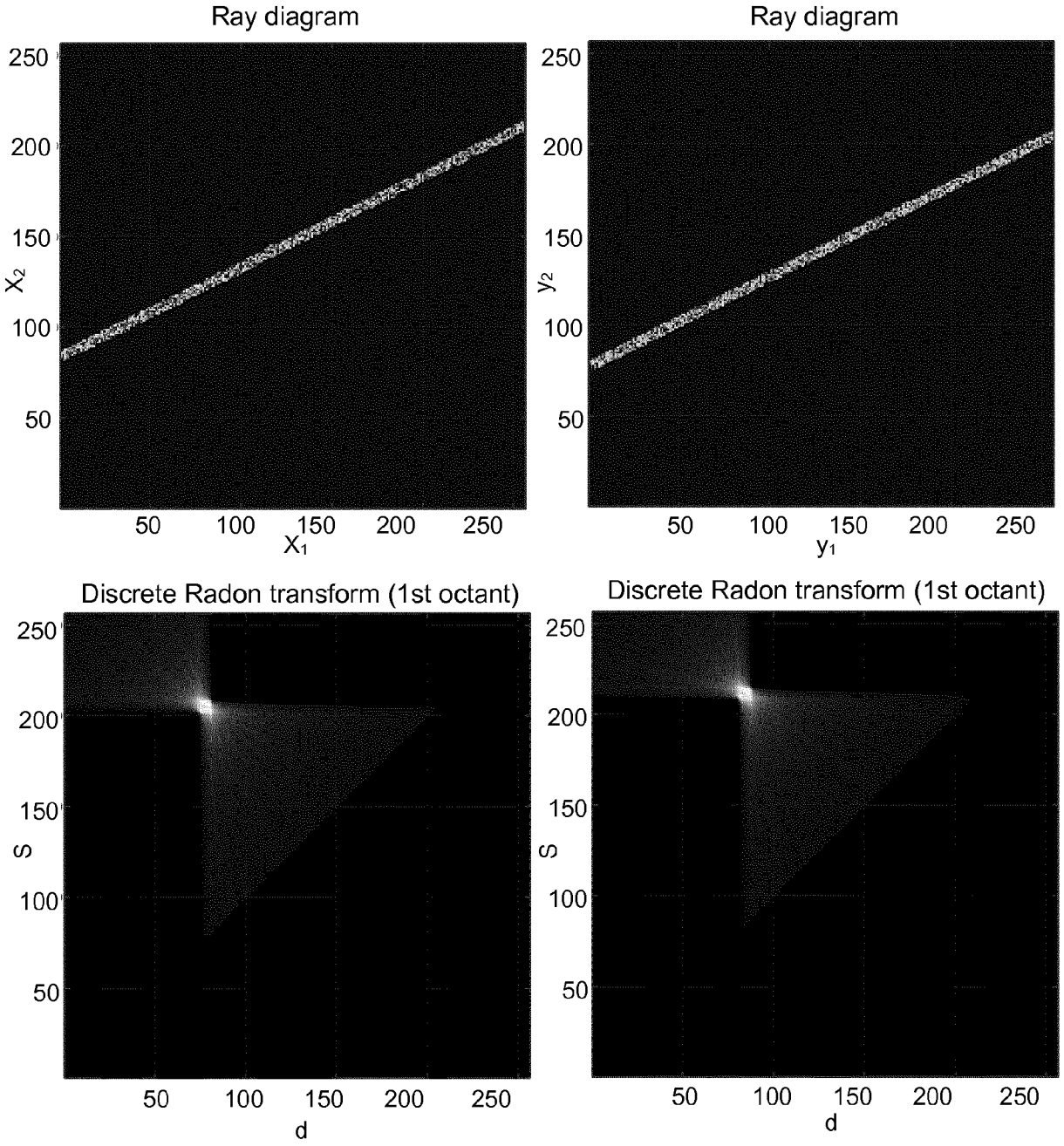
Ray diagram
250
200
150
100
50
X2
50
100
150
200
250
X1
Ray diagram
250
200
150
100
50
y2
50
100
150
200
250
y1
Discrete Radon transform (1st octant)
250
200
150
100
50
S
50
100
150
200
250
d
Discrete Radon transform (1st octant)
250
200
150
100
50
S
50
100
150
200
250
d

FIG.25

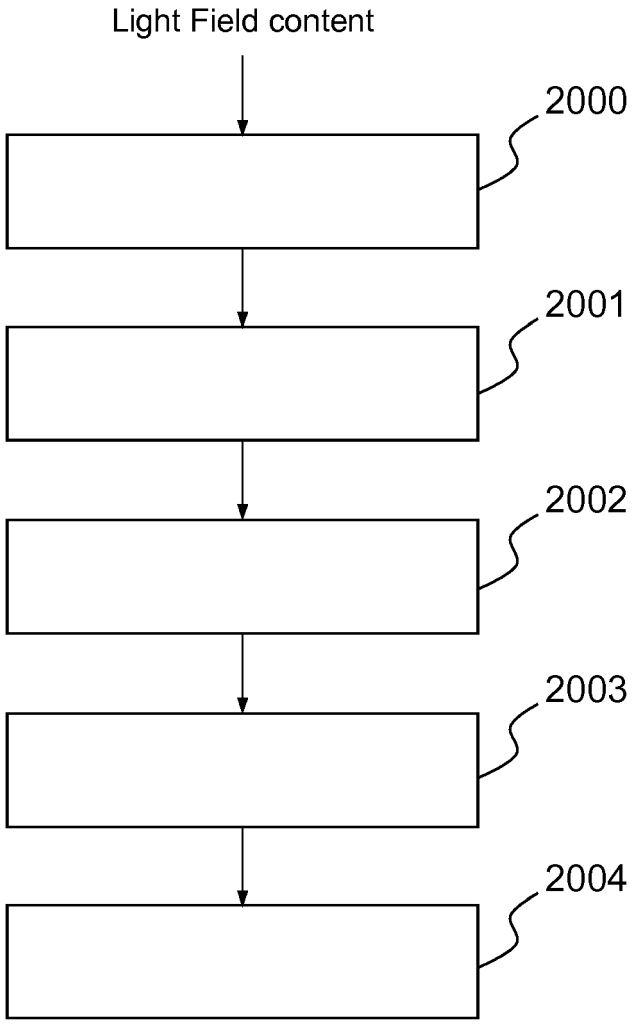

FIG.26(a)

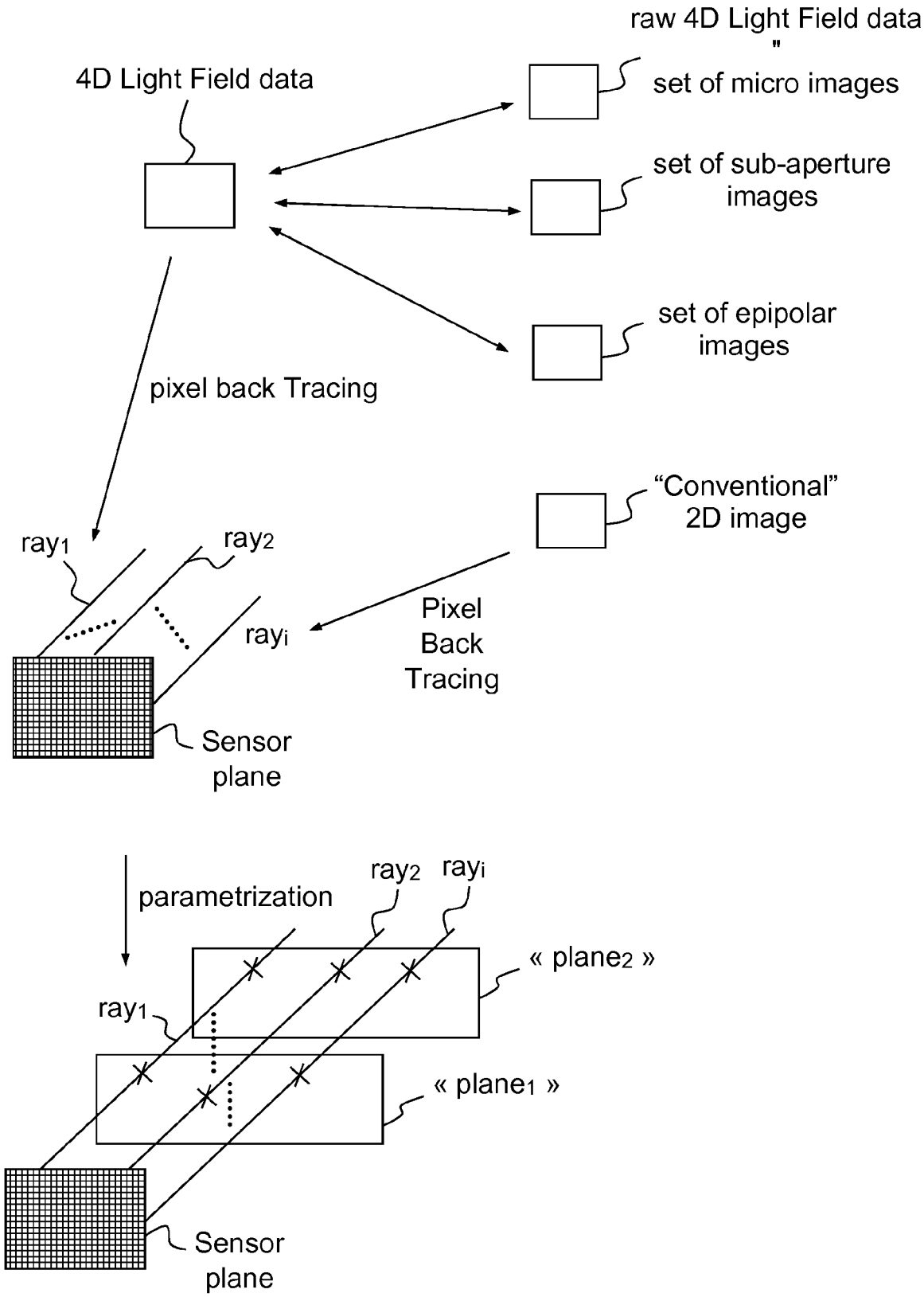
raw 4D Light Field data
"
4D Light Field data
set of micro images
set of sub-aperture images
set of epipolar images
pixel back Tracing
"Conventional" 2D image
ray1
ray2
rayi
Pixel Back Tracing
Sensor plane
parametrization
ray2
rayi
« plane2 »
ray1
« plane1 »
Sensor plane

FIG.26(b)

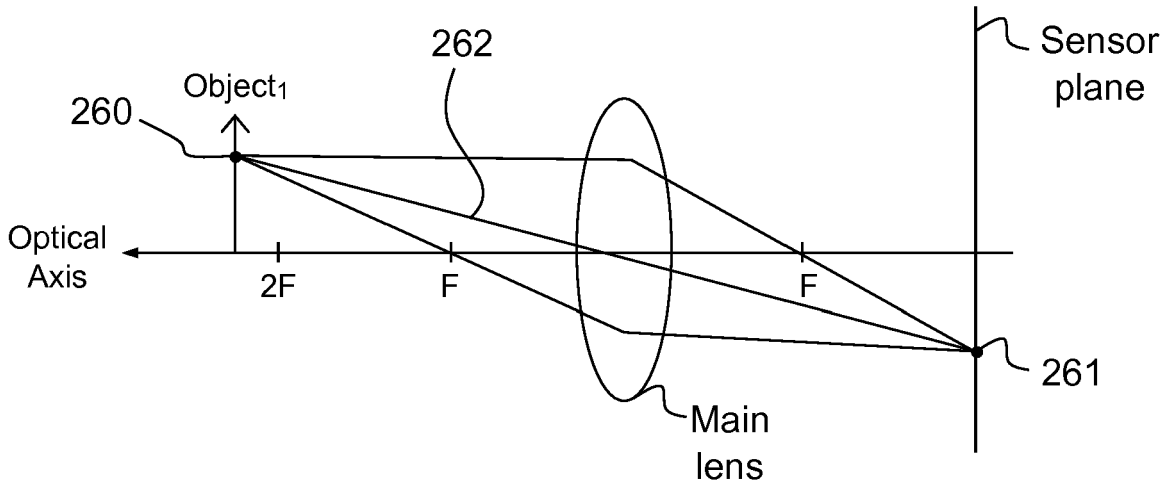

FIG.26(c)

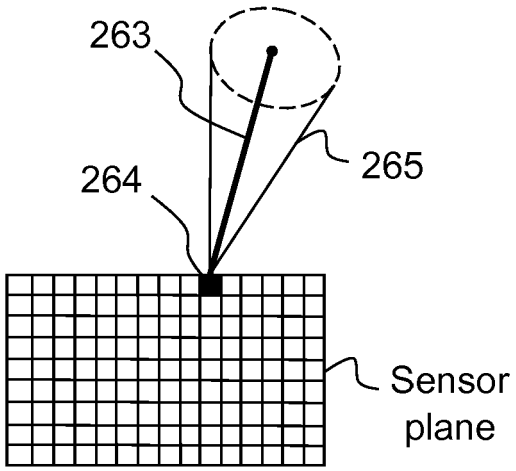

FIG.26(d)

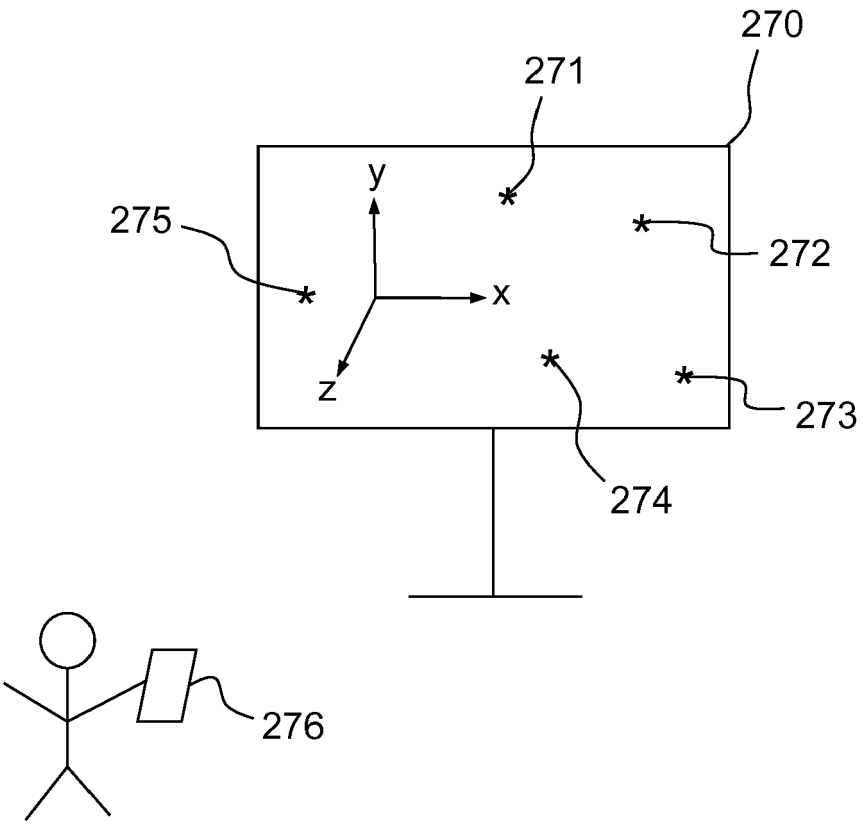

FIG.27

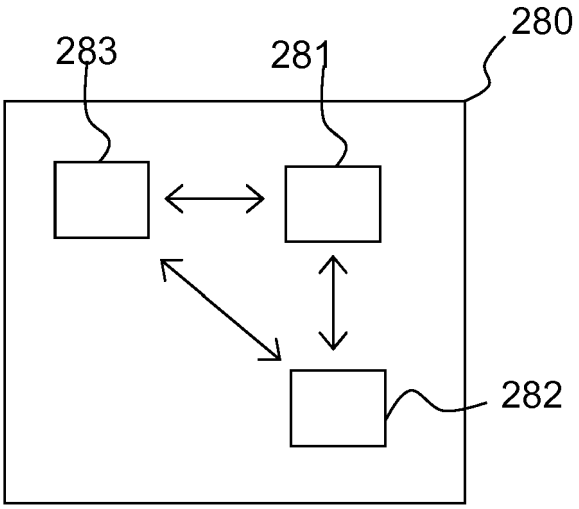

FIG.28

Europäisches Patentamt
European Patent Office
Office européen des brevets

EUROPEAN SEARCH REPORT

Application Number
EP 15 30 6438

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X,D	US 6 023 523 A (COHEN MICHAEL F [US] ET AL) 8 February 2000 (2000-02-08) * columns 4-6 * -----	1-8	INV. H04N19/597
A	US 2015/146032 A1 (RIME LAURENT [CH] ET AL) 28 May 2015 (2015-05-28) * paragraphs [0144] - [0166] * -----	1-8	
A	CHA ZHANG ET AL: "Compression of Lumigraph with multiple reference frame (MRF) prediction and just-in-time rendering", DATA COMPRESSION CONFERENCE, 2000. PROCEEDINGS. DCC 2000 SNOWBIRD, UT, USA 28-30 MARCH 2000, LOS ALAMITOS, CA, USA,IEEE COMPUT. SOC, US, 28 March 2000 (2000-03-28), pages 253-262, XP010377365, ISBN: 978-0-7695-0592-3 * Chapters 1-3 * -----	1-8	
A	VUILLEMIN J E: "Fast linear Hough transform", APPLICATION SPECIFIC ARRAY PROCESSORS, 1994. PROCEEDINGS. INTERNATIONA L CONFERENCE ON SAN FRANCISCO, CA, USA 22-24 AUG. 1994, LOS ALAMITOS, CA, USA,IEEE COMPUT. SOC, 22 August 1994 (1994-08-22), pages 1-9, XP010100274, DOI: 10.1109/ASAP.1994.331821 ISBN: 978-0-8186-6517-2 * Chapters 1-3 * -----	1-8	TECHNICAL FIELDS SEARCHED (IPC) H04N

-/--

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
Munich	11 March 2016	Lindgren, Johan

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons
.....
& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

Europäisches Patentamt
European Patent Office
Office européen des brevets

EUROPEAN SEARCH REPORT

Application Number
EP 15 30 6438

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
A	BERLEMONT S ET AL: "Detection of Curvilinear Objects in Noisy Image using Feature-Adapted Beamlet Transform", 2007 IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING 15-20 APRIL 2007 HONOLULU, HI, USA, IEEE, PISCATAWAY, NJ, USA, 15 April 2007 (2007-04-15), pages I-1225, XP031463089, ISBN: 978-1-4244-0727-9 * the whole document * -----	1-8	

TECHNICAL FIELDS SEARCHED (IPC)

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
Munich	11 March 2016	Lindgren, Johan

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons
.....
& : member of the same patent family, corresponding document

1

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 15 30 6438

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

11-03-2016

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6023523 A	08-02-2000	US 6009188 A	28-12-1999
		US 6023523 A	08-02-2000
		US 6028955 A	22-02-2000
		US 6222937 B1	24-04-2001
US 2015146032 A1	28-05-2015	NONE	

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- WO 2013180192 A **[0004] [0008] [0046]**
- GB 2488905 A **[0004]**
- US 20100265386 A **[0005] [0042]**
- US 8237708 B **[0008]**
- US 6097394 A **[0008]**
- US 6023523 A **[0008] [0011]**
- US 2013258098 A **[0041]**
- WO 2014149403 A1 **[0044]**

Non-patent literature cited in the description

- **ANAT LEVIN.** Understanding camera trade-offs through a Bayesian analysis of light field projections. *the conference proceedings of ECCV,* 2008 **[0003]**
- **J.-H. PARK et al.** Light ray field capture using focal plane sweeping and its optical reconstruction using 3D displays. *OPTICS EXPRESS,* October 2014, vol. 22 (21 **[0004]**
- **A. LEVIN.** Image and depth from a conventional camera with a coded aperture. *the proceedings of SIGGRAPH,* 2007 **[0005] [0042]**
- Extended depth of field through wave-front coding. **EDWARD R. ; DOWSKI, JR. ; W. THOMAS CATHE.** Applied Optics. 10 April 1995 **[0005]**
- **REN NG.** *Digital Light Field Photography,* July 2006 **[0006]**
- **S. WANNER.** Generating EPI Representation of a 4D Light Fields with a Single Lens Focused Plenoptic Camera. *the conference proceedings of ISVC,* 2011 **[0006]**
- **CHI-KAI LIANG.** Light Field Analysis for Modeling Image Formation. *IEEE Transactions on Image Processing,* February 2011, vol. 20 (2 **[0008]**
- **MAR LEVOY ; PAT HANRAHAN.** *Light Field Rendering* **[0008]**
- **D. LANMAN ; D. LUEBKE.** Near-Eye Light Field Displays. *the conference SIGGRAPH,* July 2013 **[0008]**
- **TOM E. BISHOP ; PAOLO FAVARO.** The Light Field Camera: Extended Depth of Field, Aliasing, and Superresolution. *the IEEE Transactions on Pattern Analysis and Machine Intelligence,* May 2012, vol. 34 (5 **[0040]**
- **EDWARD R. ; DOWSKI, JR. ; W. THOMAS CATHE.** Extended depth of field through wave-front coding. *Applied Optics,* 10 April 1995 **[0042]**
- **B. WILBURN et al.** High Performance Imaging Using Large Camera Arrays. *ACM Transactions on Graphics,* July 2005, vol. 24 (3), 765-776 **[0044]**
- **X. GU.** Polyhedral geometry and the two-plane parameterization. *Rendering Techniques'97, Proceedings of the Eurographics Workshop,* 16 June 1997 **[0082]**
- **W. A. GÖTZ ; H. J. DRUCKMÜLLER.** A fast digital radon transform-an efficient means for devaluating the hough transform. *Pattern Recogn.,* December 1995, vol. 28 (12), 1985-1992 **[0090]**
- Discrete Radon transform has an exact, fast inverse and generalizes to operations other than sums along lines. Proc. Natl. Acad. Sci. USA. W. H. Press, 2006, vol. 103, 19249-19254 **[0090]**